Mílo's Díctionary



with lexicons of programming language reserved and keywords

Mílo's Díctionary

with lexicons of computer programming language reserved and keywords

Book: Milo's Dictionary and lexicon of computer programming language reserved and key words

Author: Milo

Edition: 0.35

Cover Image Credits: Image by PDPics from Pixabay

Note that the dictionary follows the language lexicons.

Milo's Dictionary introduction

including a lexicon of computer programming language reserved and keywords

Copyright © 2021, 2022 Milo

A dictionary (including a lexicon of computer programming language reserved and keywords).

Note that the dictionary follows the language lexicons.

A little more than two decades ago, a brand new Wikipedia took at least five of my web pages and used them as their entire web page. Initially they gave my website credit, but they removed the credit after a year or two. Yes, they added to my materials, but they didn't remove my material when they took away my credit.

I am using materials from Wikipedia, but they give permission under their license. And unlike the rich people who run WIkipedia I am giving them credit, because I'm not rich, so I'm honest.

IMPORTANT NOTE:

From Help Annie GoFundMe page:

I'm Jeff, and I'd like to help a friend, Annie, who is suffering from depression and recovering from domestic abuse. She is having a tough time keeping a roof over her head, and is scheduled to be evicted on Monday or Tuesday. So I'm trying to raise money so she can move out from the motel and get into something more stable.

She's in the state of Virginia, and me and another friend of hers (Milo) are in California and are trying to raise money to, 1 - keep a roof over her head, and 2 - get her into a better place. The Virginia Rent Relief Program was going to pay her rent, but fell through at the last minute because they won't pay for a motel.

So, the most immediate problem is paying for her motel, which is \$312 a week. Then she needs help with a first and last, etc... to move into something more stable.

I've suffered from severe depression before, and I know how horrible it is. I'd love to pay for

everything, but I'm only barely getting by myself. So I'm asking for help for her.

Thank you for your time, and anything you can do to help her!

4

Permissions: Anyone may make private personal use of any of these materials.

Permissions: Professors, teachers, tutors, and other educational professionals, as well as accredited schools, may duplicate and distribute for free to students.

External links are marked like this.

Images: I am making liberal use of the royalty free artwork at Pixabay. I am including all of the artist credits (although emojis are being dropped until I figure out how to get them into a PDF).

Wikis: Initially I intend to make use of information from Wikis, including Wikipedia, to quickly get initial information into the drafts of this book, but depending on how things move forward and your feedback, I may keep some amount of this material and just credit the correct source and license.

Game copyrights: I am going to quote here from the US Copyright Office:

.....

Copyright does not protect the idea for a game, its name or title, or the method or methods for playing it. Nor does copyright protect any idea, system, method, device, or trademark material involved in developing, merchandising, or playing a game. Once a game has been made public, nothing in the copyright law prevents others from developing another game based on similar principles. Copyright protects only the particular manner of an author's expression in literary, artistic, or musical form.

Material prepared in connection with a game may be subject to copyright if it contains a sufficient amount of literary or pictorial expression. For example, the text matter describing the rules of the game or the pictorial matter appearing on the gameboard or container may be registrable.

If your game includes any written element, such as instruction or directions, the Copyright Office recommends that you apply to register it as a literary work. Doing so will allow you to register all copyrightable parts of the game, including any pictorial elements. When the copyrightable elements of the game consist predominantly of pictoral matter, you should apply to register it as a work of the visual arts.

SDRs: I am including information from SDRs released by Wizards of the Coast under one of the Open Game Licenses.

I was placing the legal notice for d20 SDR Game System License after each quotation, but it adds too much extraneous repeated info, so now it is here just once:

Legal Disclaimer

This SRD reference Requires the use of the Dungeons & Dragons Player's Handbook Third Edition, published by Wizards of the Coast, Inc.

Dungeons & Dragons(r) and the d20 system are registered trademarks of Wizards of the Coast, Inc. in the United States and other countries and are used with permission. All images and text associated therewith are the copyrighted property of Wizards of the Coast, Inc. All rights reserved.

'd20 System' and the 'd20 System' logo are trademarks of Wizards of the Coast, Inc. and are used according to the terms of the d20 Game System License version 6.0. A copy of this License can be found here and Open Gaming License.

The System Reference Document authors are: Jonathan Tweet, Monte Cook, Skip Williams, based on original material by E. Gary Gygax and Dave Arneson. It was edited and reformatted by The Bishop based on previous formats by Bombshell and the dynamic duo of Sage & Sovelior. The document has been further adapted by the <u>Dragon.ee crew</u>.

Recipe copyrights: I am going to quote here from the US Copyright Office:

How do I protect my recipe?

A mere listing of ingredients is not protected under copyright law. However, where a recipe or formula is accompanied by substantial literary expression in the form of an explanation or directions, or when there is a collection of recipes as in a cookbook, there may be a basis for copyright protection. Note that if you have secret ingredients to a recipe that you do not wish to be revealed, you should not submit your recipe for registration, because applications and deposit copies are public records.

A recipe is a statement of the ingredients and procedure required for making a dish of food. A mere listing of ingredients or contents, or a simple set of directions, is uncopyrightable. As a result, the Office cannot register recipes consisting of a set of ingredients and a process for preparing a dish. In contrast, a recipe that creatively explains or depicts how or why to perform a particular activity may be copyrightable. A registration for a recipe may cover the written description or explanation of a process that appears in the work, as well as any photographs or illustrations that are owned by the applicant. However, the registration will not cover the list of ingredients that appear in each recipe, the underlying process for making the dish, or the resulting dish itself. The registration will also not cover the

activities described in the work that are procedures, processes, or methods of operation, which are not subject to copyright protection.

Examples:

Jules Kinder submits an application to register a cookbook, *Pie in the Sky*. In the "Author Created" field of the application, Kinder asserts a claim in "text, photographs, and compilation of ingredients." Each recipe contains a list of ingredients, instructions for making a pie, and a photograph of the finished product. The claim in a "compilation of ingredients" will not be accepted because there is no copyrightable authorship in a mere listing of ingredients. Since this claim is not acceptable, the Office may communicate with Jules Kinder to limit the extent of the registration to the text and photographs only.

Paulina Neumann submits an application to register a recipe for caesar salad dressing. In the "Author Created" field, Neumann asserts a claim in "text." The work consists of a list of eleven ingredients with the following instructions: "(1) puree anchovies, garlic, Dijon, egg yolks; (2) drizzle oil in gradually to emulsify; (3) add lemon, parmesan cheese, salt, pepper, Worcester-shire and tabasco sauce." The Office will refuse registration for this work, because the list of ingredients is uncopyrightable, and the instructional text contains an insufficient amount of creative authorship.

-Library of Congress, US Copyright Office, Circular 33

.

I hope you enjoy this. I would greatly appreciate **feedback**. At the time of writing this edition, the best place is at the Twitter account <u>@TSOStheband</u> (which includes a lot of different subjects, including political views you may despise).

Even if you don't want to provide feedback and criticism of the content, you can at least point out the writing mistakes.

Note that the dictionary follows the language lexicons.

table of contents

start of dictionary
Ada lexicon
COBOL lexicon
FORTRAN lexicon
Java lexicon
JavaScript lexicon
Objective-C lexicon
PHP lexicon
PL/I lexicon
Ruby lexicon
SQL lexicon
VHDL lexicon
A a
B b
Cc
D d
E e
Ff
Gg

H h
Ii
Jj
K k
LI
M m
N n
O 0
Рр
Qq
R r
S s
T t
Uu
Vv
W w
Xx
Yy
Zz
digits 0 1 2 3 4 5 6 7 8 9
0 zero
1 one
2 two

3 three
4 four
5 five
6 six
7 seven
8 eight
9 nine
control codes
NULL
SOH (start of heading)
STX (start of text)
ETX (end of text)
EOT (end of transmission)
SEL (select)
ENQ (enquiry)
ACK (acknowledge)
RNL (require new line)
BELL
BS (backspace)
GE (graphic escape)
 HT (horizontal tabulation)
SPS (superscript)

4	^
	υ

LF (line feed) or NL (new line)
RPT (repeat)
VT (vertical tabulation)
FF (form feed)
CR (carriage return)
SO (shift out)
SI (shift in)
DLE (data link escape)
DC1 (device control 1) or XON (transmit on)
DC2 (device control 2) or TAPE
DC3 (device control 3) or XOFF (transmit off)
DC4 (device control 4) or TAPE
RES/ENP (retore, enable presentataion)
NAK (negative acknowledge)
NL (new line)
SYNC (synchronous idle)
ETB (end of transmission block)
POC (program operator control)
CAN (cancel)
EM (end of medium)
SUB (substitute)
UBS (unit backspace)

ESC (escape)

CU1 (customer use one)
FS (file separator)
GS (group separator)
RS (record separator)
US (unit separator)
IUS/ITB (intermediate transmisson block)
DS (digit select)
SOS (start of significance)
FS (field separator)
WUS (word underscore)
DEL (delete)
mathematical operators
+ (plus sign)
- (minus sign)
* (multiplication)
/ (division)
< (less than sign)
= (equals sign)
> (greater than sign)
punctuation and symbols
b (space character)
! (exclamation mark)

! (quotation mark)
(number sign)
\$ (dollar sign)
% (percent sign)
& (ampersand)
' (apostrophe)
((left parenthesis)
) (right parenthesis)
* (asterisk)
+ (plus sign)
, (comma)
- (hyphen)
. (period)
/ (forward slash)
: (colon)
; (semicolon)
< (less than sign)
= (equals sign)
> (greater than sign)
? (question mark)
(a) (commercial at sign)
[(left square bracket)

\ (backslash)

] (right square bracket)	
^ (circumflex accent)	
_(underscore)	
` (grave accent)	
{ (left curly bracket)	
(vertical line)	
<pre>} (right curly bracket)</pre>	
~ (tilde)	
μ (micro)	
Greek alphabet	
Greek alpha A α	
Greek beta B β	
gamma Γ γ	
delta Δ δ	
epsilon E ε	
zeta Z ζ	
eta Η η	
theta Θθ ϑ	
iota I ı	
карра К к	
lam(b)da Λ λ	
mu M μ	

Milo's Dictionary (with Programming Lexicons)

nu N v

xi Ξξ

omicron O o

рі П **π** ज

rho P p

sigma $\Sigma \sigma \varsigma$

tau T τ

upsilon Y υ Υ

phi Φ φ

chi X χ

psi Ψ ψ

omega $\Omega \omega$

emergency message

timeline

what if language

overview of thought experiment

history

source code

identifiers

keywords

literals

constants

white space
new lines
comments
preprocessor
declarations
types
symbols
logical types
numeric types
integer types
real types
complex types
character types
string types
relational operators and ordered types
pictures and formats
pointer types
reference and dereference
access types
dynamic storage
vector types
matrix types

a 1	rray types
ta	able types
ra	anges
e	numeration types
tı	ıples
SG	ets
st	ructure types
u	niontypes
r	ecord types with discriminants
h	ashes
li	sts
st	acks
q	ueues
d	ictionaries
0	bjects
fı	unction types
V	oid types
u	ser defined types
fi	les
SC	cope and visibility
C	onversions
e	xpressions
р	rimary expressions

postfix expressions
unary expressions
binary operator expressions
logical operator expressions
logic assertions
conditional expressions
assignment expressions
sequential expressions
constrant expressions
order of evaluation
statements
expression statements
arithmetic statements
labeled statements
blocks
compounds tatements
conditional statements
loops
interative statements and iterators
case statements
break and continue statements
return statements

goto statements
low-level statements
null statements
conditions
exceptions
errors
procedures
functions
recursion
methods
lambdas
input statements
output statements
database statements
classes
modules
packages
generic units
reflection
operations
resources
 globals

tasks
semaphores
asycnhronous message passing
remote procedure call
rendvous
virtualmachines
low-level programming
standard library
standard language additions
character processing functions
string processing functions
memory functions
input output functions
storage allocation functions
mathematical functions
time and date functions
control functions
miscellaneous functions
emergency message

Note that the dictionary follows the language lexicons.

emergency

messag

IMPORTANT NOTE:

From Help Annie GoFundMe page:

I'm Jeff, and I'd like to help a friend, Annie, who is suffering from depression and recovering from domestic abuse. She is having a tough time keeping a roof over her head, and is scheduled to be evicted on Monday or Tuesday. So I'm trying to raise money so she can move out from the motel and get into something more stable.

She's in the state of Virginia, and me and another friend of hers (Milo) are in California and are trying to raise money to, 1 - keep a roof over her head, and 2 - get her into a better place. The Virginia Rent Relief Program was going to pay her rent, but fell through at the last minute because they won't pay for a motel.

So, the most immediate problem is paying for her motel, which is \$312 a week. Then she needs help with a first and last, etc... to move into something more stable.

I've suffered from severe depression before, and I know how horrible it is. I'd love to pay for everything, but I'm only barely getting by myself. So I'm asking for help for her.

Thank you for your time, and anything you can do to help her!

~Jeff

Ada

lexícon

keywords

abort
abs
abstract
accept
access
aliased
all
and
array
at
begin

Milo's Dictionary (with Programming Lexicons)

IVIIIO	s Dictionary (with rogramming Lexicons)	22
	body	
	case	
	constant	
	declare	
	delay	
	delta	
	digits	
	do	
	else	
	elsif	
	end	
	entry	
	exception	
	exit	
	for	
	function	
	generic	
	goto	
	if	
	in	
	interface	
	is	
	limited	

Edition 0.35

mod
new
not
null
of
or
others
out
overriding
package
pragma
private
procedure
protected
raise
range
record
rem
renames
requeue
return
reverse

Milo's Dictionary (with Programming Lexicons)

select
seperate
subtype
synchronized
tagged
task
terminate
then
type
until
use
when
while
with
xor
+
-
*
/

COBOL

lexicon

COBOL reserved keywords

accept
access
add
address
advancing
after
all
alphabet
alphabetic
alphabetic-lower
alphabetic-upper

alph	anumeric
alph	anumeric-edited
also	
alter	
alter	nate
and	
any	
appl	y .
are	
area	
area	6
ascei	nding
assig	n
at	
auth	or
basis	
befo	·e
begii	nning
bina	^r y
blan	K
bloc	ζ
botto	om
by	

cancel
case
cbl
cd
cf
ch
character
characters
class
class-id
clock-units
close
cobol
code
code-set
collating
column
com-reg
 comma
common
communication
comp

comp-1

Edition 0.35

comp-2
comp-3
comp-4
comp-5
computational
computational-1
computational-2
computational-3
computational-4
computational-5
compute
configuration
contains
content
control
controls
converting
сору
corr
corresponding

count

currency

cursor
data
date-compiled
date-written
day
day-of-week
 dbcs
de
debug-contents
debug-item
debug-line
debug-name
debug-sub-1
debug-sub-2
debug-sub-3
debugging
decimal-point
declaratives
delete
delimited
delimiter
depending

descending
destination
detail
display
display-1
divide
division
down
duplicates
dynamic
egcs
egi
eject
else
emi
enable
end
end-add
end-call
end-compute
end-delete
end-divide
end-evaluate

end-if
end-invoke
end-multiply
endofdata
end-of-page
end-perform
end-read
end-receive
end-return
end-rewrite
end-search
end-start
end-string
end-subtract
end-unstring
end-write
ending
enter
entry
environment
еор
equal

error
esi
evaluate
every
exception
exit
extend
external
false
fd
file
file-control
filler
final
first
footing
for
from
function
generate
giving
global
go

goback
greater
group
heading
high-value
high-values
id
identification
if
in
index
indexed
indicate
inherits
initial
initialize
initiate
input
input-output
insert
inspect
installation

into
invalid
invoke
i-0
i-o-control
is
just
justified
kanji
key
label
last
leading
left
length
less
limit
limits
linage
linage-counter
line
line-counter
lines

linkage
local-storage
lock
low-value
low-values
memory
merge
message
metaclass
method
method-id
mode
modules
more-labels
move
multiple
multiply
native
native-binary
 negative
next
 no

not
null
nulls
number
numeric
numeric-edited
object
object-computer
occurs
of
off
omitted
on
open
optional
or
order
organization
other
output
overflow
override
packed-decimal
Edition 0.35

_	
	padding
	page
	page-counter
	password
	perform
	pf
	ph
	pic
	picture
	plus
	pointer
	position
	positive
	printing
	procedure
	procedure-pointer
	procedures
	proceed
	processing
	program
	program-id
	purge

queue
quote
quotes
random
rd
read
ready
receive
record
recording
records
recursive
redefines
reel
reference
references
relative
release
reload
rem
remainder
removal

renames

replace
replacing
report
reporting
reports
repository
rerun
reserve
reset
return
return-code
returning
reversed
rewind
rewrite
rf
rh
right
rounded
run
 same
sd

search
section
security
segment
segment-limit
select
self
send
sentence
seperate
sequence
sequential
service
set
shift-in
shift-out
sign
size
skip1
skip2
skip3
sort

sort-core-size
sort-file-size
sort-merge
sort-message
sort-mode-size
sort-return
source
source-computer
space
spaces
special-names
standard
standard-1
standard-2
start
status
stop
string
sub-queue-1
sub-queue-2
sub-queue-3
subtract

IVIIIO'S	s Dictionary (with Programming Lexicons)	42
	sum	
	super	
	suppress	
	symbolic	
	sync	
	sychronized	
	table	
	tally	
	tallying	
	tape	
	terminal	
	terminate	
	test	
	text	
	than	
	then	
	through	
	thru	
	time	
	times	
	title	
	to	

top

Edition 0.35

trace
trailing
true
type
unit
unstring
until
up
upon
usage
use
using
value
values
varying
when
when-compiled
with
words
working-storage
write
write-only

Milo's Dictionary (with Programming Lexicons)

zero

zeroes

zeros

FORTRAN

lexicon

FORTRAN reserved keywords

Portions of this document are provided here. Use the link to view and read the entire document.

NOTE: This document was released almost a year *before* the language was released.

From the <u>IBM Programmer's Reference Manual FORTRAN</u> Automatic Coding System for the IBM 704:

THE FORTRAN SYSTEM

The IBM Mathematical Formula Translating System FORTRAN is an automatic coding system for the IBM 704 EDPM. More precisely, it is a

704 program which accepts a source program written in a language — the FORTRAN language — closely resembling the ordinary language of mathematics, and which produces an object program in 704 machine language, ready to be run on a 704.

FORTRAN therefore in effect transforms the 704 into a machine with which communication can be made in a language more concise and more familiar than the 704 language itself. The result should be a considerable reduction in the training required to program, as well as in the time consumed in writing programs and eliminating their errors.





Among the features which characterize the FORTRAN system are the following:

Size of Machine Required

The system has been designed to operate on a "small" 704, but to write object programs for any 704. (For further details, see the section on Source and Object Machines in Chapter 7.) If an object program is produced which is too large for the machine on which it is to be run, the programmer must subdivide the program.

Efficiency of the Object Program

Object programs produced by FORTRAN *will nearly as efficient as those writte by good programmers.*

Scope of Applicability

The FORTRAN language is intended to be capable of expressing any problem of numerical computation. In particular, it deals easily with problems containing large sets of formulae and many variables, and it permits any variable to have up to three independent subscripts.

However, for problems in which machine words have a logical rather than a numerical meaning it is less satisfactory, and it may fail entirely to express some such problems. Nevertheless, many logical operations not directly expressible in the FORTRAN language can be obtained by making use of the provisions for incorporating library routines.

Inclusion of Library Routines

Pre-written routines to evaluate any single-valued functions of any number of arguments can be made available for incorporation into object programs by placing them on the master FORTRAN tape.

Provisions for Input and Output

Certain statements in the FORTRAN language cause the object program to be equipped with its necessary input and output programs. Those which deal with decimal information include conversion to or from binary, and permit considerable freedom of format in the external medium.

Nature of Fortran Arithmetic

Arithmetic in the object program will generally be performed with single-precision 704 floating point numbers. These numbers provide 27 binary digits (about 8 decimal digits) of precisionn, and may have magnitudes between approximately 10^{-38} and 10^{38} , and zero. Fixed point arithmetic, but for integers only, is also provided.

CHAPTER 1. GENERAL PROPERTIES OF A FORTRAN SOURCE PROGRAM

A FORTRAN source program consists of a sequnce of FORTRAN *statements*. There are 32 different types of statements, which are described in detail in the chapters which follow.

Keypunching the Program

Each statement is punched on a separate card. If a statement is too long to fit on a single card it can be continued over as many as 9 additional *continuation* cards. For each statement the initial card must contain either a zero or blank in column 6; on continuation cards column 6 must not contain a zero or blank, and it should be used to number the continuation cards consecutively from 1 to 9.

If a statment is too long to fit on a single line of the coding form, the programmer can signal to the keypuncher that he has continued on to the next line by placing a mark in the column labeled CONTINUATION.

The order of the statements is governed solely by the order of the cards. However, any number less than 2^{15} (=32768) may be associated with any statement by punching it in columns 1-5 of the initial card bearing that statemet. Thereupon this number becomes the *statement number* of that statement. Statement numbers, besides permitting cross-references within the source code, also help the programmer to correlate the object program with his source code.

Punching the character C in column 1 will cause the card to be ignored by FORTRAN. Such cards may therefore be used to carry comments which will appear when the deck is listed.

Columns 73-80 are not read by FORTRAN and may be punched with any desired identifying information.

The statements themselves are punched in columns 7-72, both on initial and continuation cards. Thus a statement consists of not more than $10 \ge 66 = 660$ characters. A table of the admissible characters in FORTRAN is given in Appendix A.

Blank characters, except in column 6, are simply ignored by FORTRAN, and the programmer may use blanks freely to improve the readability of his FORTRAN listing.

Preview of the Fortran Statements

The 32 types of statement, of which every FORTRAN program is composed, may be classified as follows:

1. The *arithmetic formula*, which causes the object program to carry out a numerical computation. Chapter 2 discusses the symbols available for referring to constants and variables, and Chapter 3 the combining of these into arithmetic formulas.

- **2.** The 15 *control statments,* which govern the flow of control in the object program. These are discussed in Chpater 4.
- **3.** The 13 *input-output statements*, which provide the object program wiht its necessary input and putput routines. These are discussed in Chapter 5.
- **4.** Finally, the 3 *specification statements,* which provide various information required or desirable to make the object program efficient, are discussed in Chapter 6.

Chapters 7 and 8, which conclude the manual, give additional detail on various topics and examples of FORTRAN programming.

Java

lexícon

keywords

abstract
assert
boolean
break
byte
case
catch
char
class
const
continue

default
do
double
else
enum
extends
final
finally
float
for
goto
if
implement
imports
instanceof
int
interface
long
native
new
package
private
protected

public
return
short
static
strictfp
super
switch
synchronized
this
throw
throws
transient
try
void
volatile
while

reserved words for literal values

false		
null		
true		

JavaScrípt lexícon

JavaScript reserved keywords

abstract	
arguments	
boolean	
break	
byte	
case	
catch	
char	
class	
const	
continue	

de	bugger
de	fault
de	lete
do	
do	uble
els	e
en	um
eva	al
exj	port
ext	tends
fal	se
fin	al
fin	ally
flo	at
for	ſ
fui	nction
go ¹	to
if	
im	plements
im	port
in	
ins	stanceof
int	

interface	
let	
long	
native	
new	
null	
package	
private	
protected	
public	
return	
short	
static	
super	
switch	
synchronized	
this	
throw	
throws	
transient	
true	
try	

Milo's Dictionary (with Programming Lexicons)

typeof
var
void
volatile
while
with
yield

JavaScript Objects, Properties, and Methods

Array
Date
eval
function
hasOwnProperty
Infinity
isFinite
isNaN
isPrototypeOf
length
Math
name
NaN
Number

55

prototype		
String		
toString		
undefined		
valueOf		

JavaScript Windows objects and properties reserved words

	alert
	all
	anchor
	anchors
	area
	assign
	blur
	button
	checkbox
	clearInterval
	clearTimeout
	clientInformation
	close
	closed
	confirm
	constructor
56	Edition 0.35

Crypto
decodeURI
decodeURIComponent
defaultStatus
Document
Element
Embed
embeds
encodeURI
encodeURIComponent
escape
Event
fileUpload
focus
forms
Frame
frames
frameRate
hidden
History
innerHeight
innerWidth

Image
link
location
navigate
Navigator
offscreenBuffering
open
opener
option
outerHeight
outerWidth
pageXOffset
pageYOffset
parent
parseFloat
parseInt
prompt
propertyIsEnum
reset
screenX
screenY
 scroll
Select

1

self	
setInterval	
setTimeout	
status	
submit	
taint	
text	
top	
unescape	
untaint	
window	

JavaScript HTML Event Handlers

	onblur	
	onclick	
	onerror	
	onfocus	
	onkeydown	
	onkeypress	
	onkeyup	
	onload	
	onmousedown	
59	onmouseover	Edition 0.35

onmouseup

onsubmit

Objective-C lexícon

keywords

atomic
auto
break
case
char
const
continue
default
do
double
else

62

enum
extern
float
for
goto
if
inline
int
long
nonatomic
register
restrict
retain
return
short
signed
sizeof
static
struct
switch
typedef
union
unsigned

void

volatile

while

_Bool

_Complex

_Imaginary

words

BOOL
bycopy
byref
Class
id
IMP
in
inout
nil
NO
NULL
oneway
out
Protocol

SEL
self
Super
YES
@catch()
@class
@dynamic
@end
@finally
@implementation
@interface
@private
@property
@protected
@protocol
@public
@selector
<pre>@selector @synthesize</pre>
<pre>@selector @synthesize @throw</pre>

PHP

lexicon

keywords

abstract
and
array()
as
break
callable
case
catch
class
clone
const

continue
declare
default
die()
do
echo
else
elseif
empty()
enddeclare
endfor
endforeach
endif
endswitch
endwhile
eval()
exit()
extends
final
finally
for
foreach
function

Edition 0.35

global
goto
if
implements
include
include_once
instanceof
insteadof
interface
isset()
list()
namespace
new
or
print
private
protected
public
require
require_once
return
static

switch		
throw		
trait		
try		
unset()		
use		
var		
while		
xor		
yield		
halt_compiler		

compile-time constants

CLASS	
DIR	
FILE	
FUNCTION	
LINE	
METHOD	
NAMESPACE	
TRAIT	

68

PL/I

lexícon

keywords

Α
ABS
ACOS
%ACT
%ACTIVATE
ADD
ADDBUFF
ADDR
ALIGNED
ALL
ALLOC

ALLOCATE
ALLOC(x)
ALLOCATION
ANY
AREA
AREA[(sixe)]
ARGn
ASCII
ASIN
ASM
ASSEMBLER
ASSOCIATE
ATAN
ATAND
ATANH
ATTENTION
ATTN
Αυτο
AUTOMATIC
В
BACKWARD
BACKWARDS

BASED

BEGIN
BINARY
BIN
BINARY(x[,p[,q]])
BIN(x[,p[,q]])
BIT
BIT(expression [size])
BKWD
BLKSIZE
BOOL
BUF
BUFFERED
BUFFERS
BUFFND
BUFNI
BUFFOFF
BUILTIN
BY
BY NAME
BYNAME
С
CALL

CALL
CEIL
CHAR
CHAR
CHARACTER
CHECK
CHECK
CLOSE
CMDCHN
COBOL — environment
COBOL — option
COLBIN
COLUMN
COMPILETIME
COMPLETION
CPLN
COMPLEX
CPLX
COMPLEX
COND
CONDITION (name)
CONDITION (attribute)
CONJG
CONN

CONNECTED
CONSECUTIVE
%CONTROL
CONTROLLED
CTL
CONV
CONVERSION
СОРУ
COS
COSD
COSH
COUNT
COUNTER
CTL360
CTLASA
CSTG
CURRENTSTORAGE
D
DATA
DATAFIELD
DATE

Milo's Dictionary (with Programming Lexicons)

DB DCL %DCL %DEACT %DEACTIVATE DEC **DEC** (function) DECIMAL **DECIMAL** (function) DECLARE %DECLARE DEFAULT DEF DEFINED DELAY DELETE **DESCRIPTION** DFT DIM DIRECT DISPLAY DIVIDE

DO

74

%DO
Ε
EDIT
ELSE
%ELSE
ЕМРТҮ
END
%END
ENDFILE
ENDPAGE
ENTRY
ENV
ENVIRONMENT
ERF
ERFC
ERROR
EVENT
EXCLUSIVE
EXIT
EXP
EXTENTNUMBER
EXT

EXTERNAL
F
F (format)
FB
FBS



keywords

alias
and
begin
BEGIN
break
case
class
def
defined?
do
else

Edition 0.35

elsif
ENCODING
end
END
ensure
false
FILE
for
if
in
LINE
module
next
nil
not
or
redo
rescue
retry
return
 self
super

then

SQL

lexícon

SQL reserved keywords

access
add
all
alter
and
and_bits
any
88
asc
audit
avg

Т

begin
between
bind
both
by
cache
call
cardinality
case
cast
char
character
char_length
char_length_b
char_length_mb
character_length
character_length_b
character_length_mb
check
cluster
column
comment
commit

compress
concatenate
cond
connect
corresponding
count
create
current
current_date
current_time
current_timestamp
cursor
data
dataxel
date
day
dbms_in_memory
dec
decimal
declare
default
delete

d	lesc
d	lirective_error
d	listinct
d	lo
d	louble
d	loublegreatethan
d	Irop
e	else
e	lseif
e	end
e	endofdata
e	eof
e	escape
e	except
e	exclusive
e	exists
f	alse
f	ïle
f	ïrst
f	loat
f	for
f	rom
f	full

Edition 0.35

grant
greateeq
group
group_by
hash
having
high_values
hintend
hint_error
hintstart
hour
identified
if
ifelse
immediate
"in
increment
index
initial
in_memory
inner
insert

i	nt
i	nteger
i	ntersect
i	nterval
i	nto
i	\$
i	terate
j	oin
l	ast
l	eading
l	eave
l	eft
l	esseq
l	evel
l	ike
l	ock
l	ong
l	oop
l	ower
l	ower_mb
l	ow_values
l	pad
r	nax

maxextents
memory_size
merge
meta_hint
min
minus
minute
mlslabel
mode
modify
month
natural
nchar
nested_loop
noaudit
no_cache
nocompress
not
noteq
nowait
null
number

Milo's Dictionary (with Programming Lexicons)

87

numeric	
octet_length	
of	
offline	
0 n	
option	
or	
or_bits	
order	
odered	
outer	
overlaps	
pctfree	
position	
position_b	
position_mb	
precision	
prior	
privileges	
public	
raw	
real	
rename	

T.

resource
revoke
 right
round
row
rowid
rownum
rows
rpad
same_source
second
select
sequentially
session
set
share
simple_nested_loop
size
smallint
start
stdout
subarray

substring_mb
substring
successful
 sum
 synonym
sysdate
table
 then
 time
 timestamp
timesten
to
 trailing
trigger
trim
trim_b
trim_mb
 true
uid
union
unique
 unknown
update

	upper
	upper_mb
	user
	using
	validate
	values
	varchar
	varchar2
	varying
-	view
	when
	whenever
	where
	with
	xor_bits
	year

VHDL

lexícon

keywords

abs
access
after
all
and
architecture
array
assert
attribute
begin
block

Т

bo	ody
bı	uffer
bı	us
Ca	all
co	omponent
co	onfiguration
co	onstant
di	isconnect
de	ownto
el	se
el	sif
er	nd
er	ntity
ex	xit
fi	le
fo)r
fu	inction
ge	enerate
ge	eneric
gı	roup
gı	uarded
if	
in	npure

in
inertial
inout
is
label
library
linkage
literal
loop
map
mod
nand
new
next
nor
not
null
of
on
open
or
others

Edition 0.35

/ilo':	s Dictionary (with Programming Lexicons)	Ś
	out	
	package	
	port	
	postponed	
	procedure	
	pure	
	range	
	record	
	register	
	reject	
	rem	
	report	
	return	
	rol	
	ror	
	select	
	severity	
	signal	
	shared	
	sla	_
	sll	
	sra	

srl

subtype
then
to
transport
type
unaffected
units
until
use
variable
wait
when
while
with
xnor
xor

start of díctionary

start of dictionary

The start of the dictionary.

Note that the dictionary follows the language lexicons.

table of contents

start of dictionary
Ada lexicon
COBOL lexicon
FORTRAN lexicon
Java lexicon
JavaScript lexicon
Objective-C lexicon

Milo's Dictionary (with Programming Lexicons)

PHP lexicon
PL/I lexicon
Ruby lexicon
SQL lexicon
VHDL lexicon
A a
B b
C c
D d
E e
F f
Gg
H h
Ii
Jj
K k
Ll
M m
N n
0 0
P p
Qq
R r

97

97

S s
T t
U u
Vv
W w
Xx
Yy
Zz
digits 0 1 2 3 4 5 6 7 8 9
0 zero
1 one
2 two
3 three
4 four
5 five
6 six
7 seven
8 eight
9 nine
control codes
 NULL
SOH (start of heading)

STX (start of text)

ETX (end of text)

EOT (end of transmission)

SEL (select)

ENQ (enquiry)

ACK (acknowledge)

RNL (require new line)

BELL

BS (backspace)

GE (graphic escape)

HT (horizontal tabulation)

SPS (superscript)

LF (line feed) or NL (new line)

RPT (repeat)

VT (vertical tabulation)

FF (form feed)

CR (carriage return)

SO (shift out)

SI (shift in)

DLE (data link escape)

DC1 (device control 1) or XON (transmit on)

DC2 (device control 2) or TAPE

DC3 (device control 3) or XOFF (transmit off)

DC4 (device control 4) or TAPE
RES/ENP (retore, enable presentataion)
NAK (negative acknowledge)
NL (new line)
SYNC (synchronous idle)
ETB (end of transmission block)
POC (program operator control)
CAN (cancel)
EM (end of medium)
SUB (substitute)
UBS (unit backspace)
ESC (escape)
CU1 (customer use one)
FS (file separator)
GS (group separator)
RS (record separator)
US (unit separator)
IUS/ITB (intermediate transmisson block)
DS (digit select)
SOS (start of significance)
FS (field separator)
WUS (word underscore)

DE	L (delete)
ma	thematical operators
+ (I	olus sign)
- (n	ninus sign)
* (n	nultiplication)
/ (d	ivision)
< (l	ess than sign)
= (e	equals sign)
> (g	greater than sign)
pur	ictuation and symbols
ħ (s	pace character)
! (e:	xclamation mark)
! (q	uotation mark)
# (n	umber sign)
\$ (d	ollar sign)
% (percent sign)
& (ampersand)
' (aj	postrophe)
((le	eft parenthesis)
) (r	ight parenthesis)
* (a	sterisk)
+ (p	olus sign)

, (comma)

- (hyphen)
. (period)
/ (forward slash)
: (colon)
; (semicolon)
< (less than sign)
= (equals sign)
> (greater than sign)
? (question mark)
@ (commercial at sign)
[(left square bracket)
\ (backslash)
] (right square bracket)
^ (circumflex accent)
_(underscore)
` (grave accent)
{ (left curly bracket)
(vertical line)
<pre>} (right curly bracket)</pre>
~ (tilde)
μ (micro)
Greek alphabet

Greek alpha A α
Greek beta B β
gamma Γ γ
delta $\Delta \delta$
epsilon Ε ε
zeta Z ζ
eta Η η
theta Θθ ϑ
iota I ı
kappa К к
lam(b)da $\Lambda \lambda$
mu M μ
nu N v
xi Ξ ξ
omicron O o
pi Π π ϖ
rho P ρ
sigma $\Sigma \sigma \varsigma$
tau T τ
upsilon Y υ Y
phi Φ φ
chi X χ
psiΨψ

omega Ω ω
emergency message
timeline
what if language
overview of thought experiment
history
source code
identifiers
keywords
literals
constants
operators and separators
white space
new lines
comments
preprocessor
declarations
types
symbols
logical types
numeric types
integer types

real types
complex types
character types
string types
relational operators and ordered types
pictures and formats
pointer types
reference and dereference
access types
dynamic storage
vector types
matrix types
array types
table types
ranges
enumeration types
tuples
sets
structure types
uniontypes
record types with discriminants
hashes
lists

stacks
queues
dictionaries
objects
function types
void types
user defined types
files
scope and visibility
conversions
expressions
primary expressions
postfix expressions
unary expressions
binary operator expressions
logical operator expressions
logic assertions
conditional expressions
assignment expressions
sequential expressions
constrant expressions
order of evaluation

statements
expression statements
arithmetic statements
labeled statements
blocks
compounds tatements
conditional statements
loops
interative statements and iterators
case statements
break and continue statements
return statements
goto statements
low-level statements
null statements
conditions
exceptions
errors
procedures
functions
recursion
methods

lambdas

input statements
output statements
database statements
classes
modules
packages
generic units
reflection
operations
resources
globals
concurrent execution
tasks
semaphores
asycnhronous message passing
remote procedure call
rendvous
virtualmachines
low-level programming
standard library
standard language additions
character processing functions
Milo's Dictionary (with Programming Lexicons)

string processing functions
memory functions
input output functions
storage allocation functions
mathematical functions
time and date functions
control functions
miscellaneous functions
emergency message

Note that the dictionary follows the language lexicons.



A

ei, letter

ASCII: 41 hex

UNICODE: LATIN CAPITAL LETTER A

keyword: PL/I format item

a

ei, letter

ASCII: 61 hex

UNICODE: LATIN SMALL LETTER A

A (w)

keyword: PL/I keyword

Format item

a1

110

programming language

Address 1 code.

Created by Matthew Newhook <matthew@engr.mun.ca> the language was used to test compiler output. It requires gcc 2.4.2 or higher and is portable to computers with memory segment protection.

ftp://ftp.cs.mun.ca/pub/a1

A+

programming language

year created: 1985 by Arthur Whitney at Morgan Stanley.

A plus

From the Wikipedia page on <u>A+ (programming language)</u>

A+ is an array programming language descended from the programming language A, which in turn was created to replace <u>APL</u> in 1988. Arthur Whitney developed the A portion of A+, while other developers at Morgan Stanley extended it, adding a graphical user interface and other language features. A+ is a high-level, interactive, interpreted language, designed for numerically intensive applications, especially those found in



financial applications. A+ runs on many Unix variants, including Linux. It is free and open source software released under a GNU General Public License.

A+ provides an extended set of functions and operators, a graphical user interface with automatic synchronizing of widgets and variables, asynchronous executing of functions associated with variables and events, dynamic loading of user compiled subroutines, and other features. A newer graphical user interface has not yet been ported to all supported platforms.

The A+ language implements the following changes to the APL language:

- an A+ function may have up to nine formal parameters
- A+ code statements are separated by semicolons, so a single statement may be divided into two or more physical lines
- The explicit result of a function or operator is the result of the last statement executed
- A+ implements an object called a dependency, which is a global variable (the dependent variable) and an associated definition that is like a function with no

111

arguments. Values can be explicitly set and referenced in exactly the same ways as for a global variable, but they can also be set through the associated definition.

Interactive A+ development is primarily done in the Xemacs editor, through extensions to the editor. Because A+ code uses the original APL symbols, displaying A+ requires a font with those special characters; a font named kapl is provided on the web site for that purpose.

Text is available under the Creative Commons Attribution-ShareAlike License

A-0 System

programming language

year created: 1951, 1952 by Grace Hopper.

From the Wikipedia page on A-0 System

A-0 system (*Arithmetic Language version 0*), written by Grace Murray Hopper in 1951 and 1952 for the UNIVAC I, was an early compiler related tool developed for electronic computers. The A-0 functioned more as a loader or linker than the modern notion of a compiler. A program was specified as a sequence of subroutines and its arguments. The subroutines were identified by a numeric code and the arguments to the subroutines were written directly after each subroutine code. The A-0 system converted the specification into machine code that could be fed into the computer a second time to execute the said program.

Text is available under the Creative Commons Attribution-ShareAlike License

See: .aac 📥 — file extension

abacus

æb'-ə-kəs (Received Pronunciation, General American), noun

plural abaci or abacuses

An instrument for performing calculations, usually a



rectangular frame with rows of wires, rods, or grooves along which beads, balls, or pebbles are slid. Each wire or groove

designates a denomination, such as units, tens, hundreds, etc (decimal system), and each bead represents a digit or a specific nuber of digits (such as representing five).

etymology: 1387 CE Middle English *abacus* from Latin: sandboard abacus, counting board, from Greek ἄβαξ (*ábax*): board covered with sand for tracing calculations, genitive form ἄβακος (*abakos*), from Phoenician, related to Hebrew אבק (*abhaq*), dust

French: l' abaque German: der abakus Russian: счеты Spanish: ábaco

A user of an abacus is called an *abacist*.

From the Wikipedia page on Abacus

The **abacus** (plural **abaci** or **abacuses**), also called a **counting frame**, is a calculating tool which has been used since ancient times. It was used in the ancient Near East, Europe, China, and Russia, centuries before the adoption of the Hindu-Arabic numeral system. The exact origin of the abacus has not yet emerged. It consists of rows of movable beads, or similar objects, strung on a wire. They represent digits. One of the two numbers



is set up, and the beads are manipulated to perform an operation such as addition, or even a square or cubic root.

In their earliest designs, the rows of beads could be loose on a flat surface or sliding in grooves. Later the beads were made to slide on rods and built into a frame, allowing faster manipulation. Abacuses are still made, often as a bamboo frame with beads sliding on wires. In the ancient world, particularly before the introduction of positional notation, abacuses were a practical calculating tool. The abacus is still used to teach the fundamentals of mathematics to some children, e.g., in post-Soviet states.

Designs such as the Japanese soroban have been used for practical calculations of up to multi-digit numbers. Any particular abacus design supports multiple methods to perform calculations, including the four basic operations and square and cube roots. Some of these methods work with non-natural numbers (numbers such as 1.5 and $\frac{3}{4}$).

Although calculators and computers are commonly used today instead of abacuses, abacuses remain in everyday use in some countries. Merchants, traders, and clerks in some parts of Eastern Europe, Russia, China, and Africa use abacuses. The abacus remains in common use as a scoring system in nonelectronic table games. Others may use an abacus due to visual impairment that prevents the use of a calculator.

Text is available under the Creative Commons Attribution-ShareAlike License

ABAP

abbreviation

programming language

From the Wikipedia page on ABAP

ABAP (Advanced Business Application Programming, originally *Allgemeiner Berichts-Aufbereitungs-Prozessor*, German for "general report preparation processor") is a high-level programming language created by the German software company SAP SE. It is currently positioned, alongside Java, as the language for programming the SAP NetWeaver Application Server, which is part of the SAP NetWeaver platform for building business applications.

Text is available under the Creative Commons Attribution-ShareAlike License

ABC

abbreviation

programming language

From the Wikipedia page on ABC (programming language)

ABC is an imperative general-purpose programming language and integrated development environment (IDE) developed at Centrum Wiskunde & Informatica (CWI), Netherlands by Leo Geurts, Lambert Meertens, and Steven Pemberton. It is interactive, structured, high-level, and intended to be used instead of BASIC, Pascal, or AWK. It is intended for teaching or prototyping, but not as a systems-programming language.

ABC had a major influence on the design of the language Python, developed by Guido van Rossum, who formerly worked for several years on the ABC system in the mid-1980s.

Features

Its designers claim that ABC programs are typically around a quarter the size of the equivalent Pascal or C programs, and more readable. Key features include:

- Only five basic data types
- No required variable declarations
- Explicit support for top-down programming
- Statement nesting is indicated by indentation, via the off-side rule
- Infinite precision arithmetic, unlimited-sized lists and strings, and other features supporting orthogonality and ease of use by novices

ABC was originally a monolithic implementation, leading to an inability to adapt to new requirements, such as creating a graphical user interface (GUI). ABC could not directly access the underlying file system and operating system.

The full ABC system includes a programming environment with a structure editor (syntax-directed editor), suggestions, static variables (persistent), and multiple workspaces, and is available as an interpreter-compiler. As of 2020, the latest version is 1.05.02, and it is ported to Unix, DOS, Atari, and Apple MacOS.

Text is available under the Creative Commons Attribution-ShareAlike License

ABC ALGOL

programming language

From the Wikipedia page on ABC ALGOL

ABC ALGOL is an extension of the programming language ALGOL 60 with arbitrary data structures and user-defined operators, intended for computer algebra (symbolic mathematics). Despite its advances, it was never used as widely as Algol proper.

Text is available under the Creative Commons Attribution-ShareAlike License

ABD(x)

I

keyword: PL/I keyword

Built-in function

abort

ə'-bo:t (Received Pronunciation), ə'-bo.t (General American), verb

keyword: Ada reserved key word

The **abort** keyword is used to abort either a task (thread) or partition (process). An abort_statement causes one or more tasks to become abnormal, thus preventing any further interaction with such tasks. The completion of the triggering_statement of an asynchronous_select causes a sequence_of_statements to be aborted. — <u>Ada Reference</u> <u>Manual 9.8</u>

abort_statement ::= **abort** *task_*name {, *task_*name}; — <u>Ada Reference Manual 9.8</u>

See: .abr 🎦 — file extension

abs

keyword: Ada reserved key word

keyword: ABS (all caps) BASIC function

keyword: VHDL reserved key word

abstract

keyword: Ada reserved key word

keyword: Java reserved key word

keyword: JavaScript reserved word

keyword: PHP reserved word

ACC

abbreviation programming language

Accent

programming language

Rational Synergy

accept

keyword: Ada reserved key word

keyword: COBOL reserved word (z/OS)

access

keyword: Ada reserved key word

keyword: COBOL reserved word (z/OS)

keyword: SQL reserved word (Oracle)

keyword: VHDL reserved key word

Ace DASL

programming language

Distributed Application Specification Language

ACOS(x)

keyword: PL/I keyword

Built-in function

acs

keyword: ACS (all caps) BASIC function

Action!

programming language

ActionScript

programming language

Actor

programming language

acute

ə'-kju:t (Received Pronunciation), ə-kyōōt' (General American), adjective

1. geometry An angle numerically smaller than a right angle, as in acute angle.

2. *geometry* Of a triangle, having all three interior angles measuring less than 90 degrees each, as in **acute triangle**.

Ada

programming language

See: Ada lexicon

add

keyword: COBOL reserved word (z/OS)

keyword: SQL reserved word (ANSI, Oracle)

ADD(x,y,p[,q]

keyword: PL/I keyword

Built-in function

ADDBUFF (n)

keyword: PL/I keyword

Option of ENVIRONMENT attribute

adder

119

æ'-dæ, noun

In a computing machine, an arithmetic component that performs the addition of positive numbers, usually integers (floating point adders exist). N.B. An arithmetic component that both adds and subtracts is called an **algebraic adder**.

ADDR(x)

keyword: PL/I keyword

Built-in function

Adenine

programming language

Haystack

address

keyword: COBOL reserved word (z/OS)

AdvPL

abbreviation

programming language

advancing

keyword: COBOL reserved word (z/OS)

a.e.

abbreviation

mathematics: Abbreviation for almost everywhere.

after

keyword: COBOL reserved word (z/OS)

keyword: VHDL reserved key word

aftertouch

keyword: MIDI comamnd A0 hex (where n is the channel number) 2 paraaemters paramater 1: key paramater 2: velocity

Agda

programming language

AgilentVEE

programming language

Keysight VEE

See: .ai 🔂 — file extension

Milo's Dictionary (with Programming Lexicons)

See: .aif 🗿

- file extension

See: .aiff 🗟 — file extension

AIMMS

abbreviation

programming language

AIX

abbreviation

operating system

Proprietary OS by IBM.

IBM's Advavanced Interactive eXecutive, a System V Unix version. —<u>Wikipedia: List of operating</u> systems

Aldor

programming language

Alef

programming language

alert

keyword: JavaScript windows method

ALF

abbreviation

programming language

algebraic adder

al'-ge-bra-ic ad'-der, noun

In a computing machine, an arithmetic component that performs both addition and subtraction.

ALGOL 58

abbreviation

programming language

ALGOL 60

abbreviation

programming language

ALGOL 68

abbreviation

programming language

ALGOL W

abbreviation 123

programming language

alias

keyword: Ruby reserved word

aliased

keyword: Ada reserved key word

Alice

programming language

Alice ML

AliceBlue

keyword: HTML color name

AliceBlue #F0F8FF

ALIGNED

keyword: PL/I keyword

Attribute

all

Milo's Dictionary (with Programming Lexicons)

keyword: Ada reserved key word

keyword: COBOL reserved word (z/OS)

keyword: JavaScript windows method

keyword: PL/C keyword

Diagnostic option of PUT statement as in PUT ALL;

keyword: SQL reserved word (ANSI, Oracle, z/OS)

keyword: VHDL reserved key word

ALL(x)

keyword: PL/I keyword

Built-in function

See: ALLOC — PL/I

See: ALLOC(x) — PL/I

ALLOCATE

keyword: PL/I keyword

Abbreviated: ALLOC

Statement

ALLOCATON(x)

keyword: PL/I keyword

Abbreviated: ALLOC(x)

Built-in function

Alma-0

programming language

alphabet

keyword: COBOL reserved word (z/OS)

alphabetic

keyword: COBOL reserved word (z/OS)

alphabetic-lower

keyword: COBOL reserved word (z/OS)

alphabetic-upper

keyword: COBOL reserved word (z/OS)

alphanumeric

keyword: COBOL reserved word (z/OS)

alphanumeric-edited

keyword: COBOL reserved word (z/OS)

also

keyword: COBOL reserved word (z/OS)

alter

keyword: COBOL reserved word (z/OS)

keyword: SQL reserved word (ANSI, Oracle)

alternate

keyword: COBOL reserved word (z/OS)

AmbientTalk

programming language

Amiga E

programming language

AmigaOS

operating system

Proprietary OS by Amiga Inc. and Commodore International.

AmigaOS 1.0-3.9 (Motorola 68000) - Wikipedia: List of operating systems

AmigaOS 4 (PowerPC) - Wikipedia: List of operating systems

Amiga Unix

operating system

Also called Amix

Proprietary OS by Amiga Inc.

Amix

abbreviation operating system Alternate name for Amiga Unix Proprietary OS by Amiga Inc.

AMOS

abbreviation

programming language

AMOS BASIC

AMPL

abbreviation

programming language

See: .amr 🝳 — file extension

AMSDOS

abbreviation

operating system

Proprietary OS by Amstrad

Analitik

programming language

anchor

keyword: JavaScript windows method

anchors

keyword: JavaScript windows method

and

conjunction: connecting conjunction

keyword: Ada reserved key word

keyword: AND (all caps) BASIC statement

keyword: COBOL reserved word (z/OS)

keyword: PHP reserved word

keyword: Ruby reserved word

keyword: SQL reserved word (ANSI, Oracle, z/OS)

keyword: VHDL reserved key word

and_bits

keyword: SQL reserved word (z/OS)

Android

operating system

Proprietary OS by Google

Android is an operating system for mobile devices. It consists of Android Runtime (userland) with Linux (kernel), with its Linux kernel modified to add drivers for mobile device hardware and to remove unused Vanilla Linux drivers. —<u>Wikipedia: List of operating systems</u>

AntiqueWhite

A.NET

abbreviation

programming language

A#/A sharp

AngelScript

programming language

See: .ani 🌌 — file extension

AntiqueWhite

keyword: HTML color name

130

any

keyword: COBOL reserved word (z/OS)

keyword: SQL reserved word (ANSI, Oracle)

ANY(x)

keyword: PL/I keyword

Built-in function

AOS

abbreviation

operating system

Proprietary OS by Data General.

For 16-bit Data General Eclipse computers. - Wikipedia: List of operating systems

AOS/VS

abbreviation

operating system

Proprietary OS by Data General.

For 32-bit Data General (MV series) Eclipses. - Wikipedia: List of operating systems

Apache Pig Latin

See: .ape - file extension

Apex

abbreviation

programming language

Salesforce.com, Inc

APL

abbreviation

programming language

apothem

ap'-o-them, noun

The perpendicular distance from the center of a regular polygon to a side.

Synonym short radius

append

keyword: APPEND (all caps) BASIC command

App Inventor

programming language

Milo's Dictionary (with Programming Lexicons) App Inventro for Android's visual block language MIT App Inventor

Apple DOS

operating system

Proprietary OS by Apple for the Apple II family

programming language

Apple Lisa

operating system

Proprietary OS by Apple for the Apple Lisa

Apple Macintosh

operating system

Proprietary OS by Apple for the Apple Macintosh.

Variants include Classic Mac OS, A/UX, Copland, MkLinux, Pink, Rhapsody, Mac OS X, OS X, macOS, Mac OS X Server, OS X Server, and macOS Server.

Apple Network Server

operating system

Proprietary OS: Apple customized version of IBM AIX.

AIX

Apple Pascal

operating system

Proprietary OS by Apple for the Apple II family

AppleScript

programming language

Apple SOS

operating system

Proprietary OS by Apple for the Apple III

apply

keyword: COBOL reserved word (z/OS)

APT

abbreviation

programming language

Aqua

keyword: HTML color name



134

#00FFFF

Aquamarine

keyword: HTML color name



Arc

programming language

ARC Assembly

programming language

year created: 1947

created by: Kathleen Booth

predecessors: ENIAC coding system

architecture

keyword: VHDL reserved key word

are

keyword: COBOL reserved word (z/OS)

area

keyword: COBOL reserved word (z/OS)

keyword: JavaScript windows object

AREA

keyword: PL/I keyword

Condition

AREA[(size)]

keyword: PL/I keyword

Attribute

areas

keyword: COBOL reserved word (z/OS)

ARexx

abbreviation

programming language

See: .arf — file extension

ARGn

keyword: PL/I keyword

Option of NOMAP, NOMAPIN, and NOMAPOUT options of the OPTIONS attribute

arguments

keyword: JavaScript reserved word

Argus

programming language

See: .arj — file extension

arm

arm, noun

A side of an angle, as in **arms of an angle**.

AROS Research Operating System

operating system

Proprietary OS by Commodore International.

A/ROSE

abbreviation

operating system

Proprietary embedded operating system by Apple.

array

keyword: Ada reserved key word

keyword: JavaScript object

keyword: PL/C keyword

Diagnostic option of PUT statement as in PUT ARRAY;

keyword: VHDL reserved key word

array()

keyword: PHP reserved word (this is a reserved word and not a function)

Arthur

operating system

Proprietary OS by Acorn Computers

See: .arw — file extension

ARX

abbreviation

operating system

Proprietary OS by Acorn Computers

as

keyword: PHP reserved word

keyword: SQL reserved word (ANSI, Oracle, z/OS)

asc

keyword: ASC (all caps) BASIC function

keyword: SQL reserved word (ANSI, Oracle, z/OS)

ascending

keyword: COBOL reserved word (z/OS)

ASCII

keyword: PL/I keyword

Option of Environment attribute

See: .asf 🗟 — file extension

See: .asm — file extension

ASIN(x)

keyword: PL/I keyword

Built-in function

See: ASM — PL/I

ASSEMBLER

keyword: PL/I keyword ¹³⁹ Abbreviated: ASM Option of OPTIONS attribute/option

Assembly language

programming language

ASM

assert

keyword: Java reserved key word

keyword: VHDL reserved key word

assign

keyword: COBOL reserved word (z/OS)

keyword: JavaScript windows method

ASSCOIATE

keyword: PL/I keyword

Option of ENVIROMENT attribute

astroid

æ'-st.joid, noun

The hypocycloid of four cusps.

See: .asx 🗟 — file extension

at

keyword: Ada reserved key word

keyword: COBOL reserved word (z/OS)

ATAN(x[,y]

keyword: PL/I keyword

Built-in function

ATAND(x[,y])

keyword: PL/I keyword

Built-in function

ATANH(x)

keyword: PL/I keyword

Built-in function

Atari DOS

operating system

Proprietary OS by Atari for 8-bit computers.

Atari MultiTOS

operating system

Proprietary OS by Atari.

Atari TOS

operating system

Proprietary OS by Atari.

atomic

keyword: Objective-C word

See: ATTN — PL/I

ATTENTION

keyword: PL/I keyword

Abbreviated: ATTN

Condition

attribute

keyword: VHDL reserved key word

audit

keyword: SQL reserved word (Oracle)

author

keyword: COBOL reserved word (z/OS)

auto

keyword: Objective-C reserved word

See: AUTO — PL/I

AutoIt

programming language

AutoLisp

programming language

Visual LISP

AUTOMATIC

keyword: PL/I keyword

Abbreviated: AUTO

Attribute

A/UX

abbreviation

operating system

Proprietary OS by Apple for the Apple Macintosh.

UNIX System V with BSD extensions. - Wikipedia: List of operating systems

Apple Macintosh

Averest

programming language

avg

keyword: SQL reserved word (z/OS)

See: .avi 📕 — file extension

AWK

abbreviation

programming language

Axum

programming language

ayacc

abbreviation

programming language
66

ayacc - UC Irvine. written in Ada, produces Ada output ftp://liege.ics.uci.edu/pub/irus/aflex-ayacc_1.2a.tar.Z

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

yacc

Azure

keyword: HTML color name



Bb

B

ASCII: 42 hex

UNICODE: LATIN CAPITAL LETTER B

b

ASCII: 62 hex

UNICODE: LATIN SMALL LETTER B

B

programming language

B [(x)]

keyword: PL/I keyword

Format item



See: .bak — file extension

BACKWARD

keyword: PL/I keyword

Abbreviated: BKWD

Option of ENVIRONMENT attribute

BACKWARDS

keyword: PL/I keyword

Attribute

Ballerina

programming language

See: .bas — file extension

BASED[(location-espression)]

keyword: PL/I keyword

Attribute

BASH

abbreviation

BASIC

abbreviation

programming language

basis

keyword: COBOL reserved word (z/OS)

See: .bar — file extension

Batch-11/DOS-11

operating system

Proprietary OS by Digital Equipment Corporation (DEC). Later owned by Compaq, Hewlett-Packard, and Hewlett Packard Enterprise

Batch file

programming language

Windows/MS-DOS

bc

abbreviation

programming language

Compression: basic calculator 148

BCPL

abbreviation

programming language

BeanShell

programming language

before

keyword: COBOL reserved word (z/OS)

begin

keyword: Ada reserved key word

keyword: PL/I keyword

Statement

keyword: BEGIN (all caps) Ruby reserved word

keyword: begin (all lower case) Ruby reserved word

keyword: SQL reserved word (z/OS)

keyword: VHDL reserved key word

beginning

BeIA

abbreviation

operating system

Proprietary OS by Be Inc.

Beige

keyword: HTML color name



BeOS

abbreviation

operating system

Proprietary OS by Be Inc.

BeOS 55.1d0

abbreviation

operating system

Proprietary OS by Be Inc.

BESYS

abbreviation

operating system

Proprietary OS by IBM

for the IBM 7090 - Wikipedia: List of operating systems

BESYS

abbreviation

operating system

Proprietary OS by Bell Labs.

BETA

programming language

between

keyword: SQL reserved word (ANSI, Oracle, z/OS)

See: .bik 🎇 — file extension

See: BIN — PL/I attribute

See: BIN — PL/I function

See: .bin — file extension

binary

keyword: COBOL reserved word (z/OS) ¹⁵¹

BINARY

keyword: PL/I keyword

Abbreviated: BIN

Attribute

BINARY(x[,p,[q]])

keyword: PL/I keyword

Abbreviated: BIN

Built-in function

bind

keyword: SQL reserved word (z/OS)

Bison

programming language

Bison - from GNU ftp://prep.ai.mit.edu/pub/gnu/bison-1.21.tar.Z

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

yacc

Bison++

programming language

Bison++ - produces C++ output. ftp://psuvax1.cs.psu.edu/pub/src/gnu/bison++-1.04.tar.Z

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

yacc

Bisque

keyword: HTML color name



BIT (length)

keyword: PL/I keyword

Attribute

BIT(expression [size])

keyword: PL/I keyword

Built-in function

See: BKWD — PL/I

Black

keyword: HTML color name



BlanchedAlmond

keyword: HTML color name

BlanchedAlmond

#FFEBCD

blank

keyword: COBOL reserved word (z/OS)

BLISS

abbreviation

programming language

BLKSIZE (expression)

keyword: PL/I keyword

Option of ENVIRONMENT attribute

block

keyword: COBOL reserved word (z/OS)

keyword: VHDL reserved key word

Blockly

programming language

BlooP

abbreviation

programming language

Blue

keyword: HTML color name



BlueViolet

keyword: HTML color name



blur

keyword: JavaScript windows method

See: .bmp 鬬 — file extension

body

keyword: Ada reserved key word

keyword: VHDL reserved key word

BOOL

keyword: Objective-C word

BOOL(x, y, z)

keyword: PL/I keyword

Built-in function

See: _Bool — C

boolean

keyword: Java reserved key word

keyword: JavaScript reserved word

Boomerang

Bosque

programming language

both

keyword: SQL reserved word (z/OS)

bottom

keyword: COBOL reserved word (z/OS)

break

keyword: Java reserved key word

keyword: JavaScript reserved word

keyword: Objective-C reserved word

keyword: PHP reserved word

keyword: Ruby reserved word

bridgeOS

operating system

Proprietary embedded operating system by Apple.

Brief Code

programming language

year created: 1949

created by: John Mauchly and William F. Schmitt

predecessors: ENIAC Short Code

Brown

keyword: HTML color name



See: BUF — PL/I

buffer

keyword: VHDL reserved key word

BUFFERED

keyword: PL/I keyword

Abbreviated: BUF

Attribute

BUFFERS (n)

keyword: PL/I keyword

Option of ENVIRONMENT attribute 158

BUFFND (n)

keyword: PL/I keyword

Option of ENVIRONMENT attribute

BUFNI (n)

keyword: PL/I keyword

Option of ENVIRONMENT attribute

BUFFOFF

keyword: PL/I keyword

Option of ENVIRONMENT attribute

BUILTIN

keyword: PL/I keyword

Attribute

BurlyWood

keyword: HTML color name



Burroughs MCP

abbreviation

operating system

Proprietary OS by Burroughs Corporation.

bus

keyword: VHDL reserved key word

button

keyword: JavaScript windows object

by

keyword: COBOL reserved word (z/OS)

keyword: SQL reserved word (ANSI, Oracle, z/OS)

BY

keyword: PL/I keyword

Option of DO statement

bycopy

keyword: Objective-C former reserved word

160

See: BYNAME — PL/I

BY NAME

keyword: PL/I keyword

Abbreviated: BYNAME

Option of assignment statement

byref

keyword: Objective-C former reserved word

byte

keyword: Java reserved key word

keyword: JavaScript reserved word

Сc

C

ASCII: 43 hex

UNICODE: LATIN CAPITAL LETTER C

C

ASCII: 63 hex

UNICODE: LATIN SMALL LETTER C

C (x[,y])

keyword: PL/I keyword

Format item

cache

keyword: SQL reserved word (z/OS)

CadetBlue

162



Caldera DR-DOS 7.02 and higher

operating system

Proprietary OS by Digital Research, Inc.

Derived from DR-DOS 3.31-6.0 - Wikipedia: List of operating systems

DOS

DR-DOS

Caldera OpenDOS 7.01

operating system

Proprietary OS by Digital Research, Inc.

Derived from DR-DOS 3.31-6.0 - Wikipedia: List of operating systems

DOS

DR-DOS

call

keyword: SQL reserved word (z/OS)

keyword: VHDL reserved key word

CALL

keyword: PL/I keyword

Statement

CALL

keyword: PL/I keyword

Option of INITIAL attribute

callable

keyword: PHP reserved word

cancel

keyword: COBOL reserved word (z/OS)

cardinailty

keyword: SQL reserved word (z/OS)

case

keyword: Ada reserved key word

keyword: COBOL reserved word (z/OS)

keyword: Java reserved key word

keyword: JavaScript reserved word ¹⁶⁴

keyword: Objective-C reserved word

keyword: PHP reserved word

keyword: Ruby reserved word

keyword: SQL reserved word (z/OS)

cast

keyword: SQL reserved word (z/OS)

catch

keyword: Java reserved key word

keyword: JavaScript reserved word

keyword: PHP reserved word

See: @catch() — Objective C

CB Unix

abbreviation

operating system

Proprietary OS by Bell Labs.

A version of UNIX Time-Sharing System v6. -Wikipedia: List of operating systems

UNIX Time-Sharing System v6

cbl

keyword: COBOL reserved word (z/OS)

CCI Multiuser DOS

abbreviation

operating system

Proprietary OS by Digital Research, Inc.

DOS

cd

keyword: COBOL reserved word (z/OS)

See: .cda 🕑 — file extension

See: .cdr 🕅 — file extension

See: .cdt 💟 — file extension

CEIL(x)

keyword: PL/I keyword

Built-in function

cf

ch

keyword: COBOL reserved word (z/OS)

channel pressure

keyword: MIDI comamnd

Dn hex (where n is the channel number)

1 paraaemter

paramater: pressure

char

keyword: Java reserved key word

keyword: JavaScript reserved word

keyword: Objective-C reserved word

keyword: SQL reserved word (ANSI, Oracle, z/OS)

See: CHAR — PL/I

CHAR(expression [size])

keyword: PL/I keyword

Built-in function

character

keyword: COBOL reserved word (z/OS)

keyword: SQL reserved word (z/OS)

167

168

CHARACTER [(lenght)]

keyword: PL/I keyword

Abbreviated: CHAR

Attribute

characters

keyword: COBOL reserved word (z/OS)

char_length

keyword: SQL reserved word (z/OS)

char_length_b

keyword: SQL reserved word (z/OS)

char_length_mb

keyword: SQL reserved word (z/OS)

character_length

keyword: SQL reserved word (z/OS)

character_length_b

168

keyword: SQL reserved word (z/OS)

character_length_mb

keyword: SQL reserved word (z/OS)

Chartreuse

keyword: HTML color name



check

keyword: PL/C keyword

Statement as in CHECK (15, 20);

keyword: SQL reserved word (ANSI, Oracle, z/OS)

CHECK [(name list)]

keyword: PL/I keyword

Condition

CHECK

keyword: PL/I keyword

Statement 169

checkbox

keyword: JavaScript windows object

Chippewa Operating System (COS)

operating system

Proprietary OS by Control Data Corporation.

Chocolate

keyword: HTML color name



Chrome OS

operating system

Proprietary OS by Google

Chrome OS is designed to work exclusively with web applications. Announced on July 7, 2009, Chrome OS is currently publicly available and was released summer 2011. The Chrome OS source code was released on November 19, 2009, under the BSD license as Chromium OS. —<u>Wikipedia: List</u> of operating systems

Chromium OS

operating system

Proprietary OS by Google

Chromium OS is an open source operating system development version of Chrome OS. Both operating systems are based on the Linux kernel. —<u>Wikipedia: List of operating systems</u>

class

keyword: COBOL reserved word (z/OS)

keyword: Java reserved key word

keyword: JavaScript reserved word

keyword: Class (initial cap) Objective-C word

keyword: PHP reserved word

keyword: Ruby reserved word

See: @class — Objective C

See: _CLASS_ — PHP

classic Mac OS

operating system

Proprietary OS by Apple for the Apple Macintosh.

Apple Macintosh

class-id

clearInterval

keyword: JavaScript windows method

clearTimeout

keyword: JavaScript windows method

clientInformation

keyword: JavaScript windows object

clock-units

keyword: COBOL reserved word (z/OS)

clone

keyword: PHP reserved word

close

keyword: COBOL reserved word (z/OS)

keyword: JavaScript windows method

CLOSE

keyword: PL/I keyword

Statement 172

closed

keyword: JavaScript windows property

cluster

keyword: SQL reserved word (Oracle)

CMDCHN

keyword: PL/I keyword

Option of ENVIRONMENT attribute

cobol

abbreviation

keyword: COBOL reserved word (z/OS)

COBOL

keyword: PL/I keyword

Option of ENVIROMENT attribute

COBOL

keyword: PL/I keyword

Option of OPTIONS attribute

code

keyword: COBOL reserved word (z/OS)

code-set

keyword: COBOL reserved word (z/OS)

COLBIN

keyword: PL/I keyword

Option of ENVIRONMENT attribute

collating

keyword: COBOL reserved word (z/OS)

column

keyword: COBOL reserved word (z/OS)

keyword: SQL reserved word (Oracle)

COLUMN (w)

keyword: PL/I keyword

Format item

See: .com — file extension

comment

keyword: SQL reserved word (Oracle, z/OS)

commit

keyword: SQL reserved word (z/OS)

Compatitble Time-Sharing System (CTSS)

operating system

Proprietary OS by IBM

developed at MIT's Computation Center for use on a modified IBM 7094 — Wikipedia: List of operating systems

COMPILETIME

keyword: PL/I keyword

Built-in function

COMPLETION(event)

keyword: PL/I keyword

Abbreviated: CPLN

Built-in function, pseudo-variable

component

keyword: VHDL reserved key word

COMPLEX

keyword: PL/I keyword

Abbreviated: CPLX

Attribute

COMPLEX(a, b)

keyword: PL/I keyword

Built-in function, pseudovariable

compress

keyword: SQL reserved word (Oracle)

com-reg

keyword: COBOL reserved word (z/OS)

comma

keyword: COBOL reserved word (z/OS)

common

communication

keyword: COBOL reserved word (z/OS)

comp

keyword: COBOL reserved word (z/OS)

comp-1

keyword: COBOL reserved word (z/OS)

comp-2

keyword: COBOL reserved word (z/OS)

comp-3

keyword: COBOL reserved word (z/OS)

comp-4

keyword: COBOL reserved word (z/OS)

comp-5

See: _Complex — Objective C

computational

keyword: COBOL reserved word (z/OS)

computational-1

keyword: COBOL reserved word (z/OS)

computational-2

keyword: COBOL reserved word (z/OS)

computational-3

keyword: COBOL reserved word (z/OS)

computational-4

keyword: COBOL reserved word (z/OS)

computational-5

keyword: COBOL reserved word (z/OS)

compute

computer

kəm'-pju:-tə (Received Pronunciation), kəm'-pju:-tə (General American), noun

1. (now rare, chiefly historical) a person employed to perform computations.

Historically, a male is called a computer or computist and a female is called a comptress.

2. A programmable device that accepts data from input and/or memory, performs prescribed mathematical calculations and logical operations, and outputs, displays, or stores the results of these computations. Most commonly an electronic digital computer.

Some examples include super computers, mainframes, minicomputers, microcomputers, desktop computers, laptop computers, tablets, smartphones, and embedded systems.

etymology: 1600-50 CE Modern English *compute* + *-er*, related to Middle French *computeur*; first known use (for an occupation) in the 1613 book *The Yong Mans Gleanings* by Richard Brathwait: "I haue [sic] read the truest computer of Times, and the best Arithmetician that euer [sic] breathed, and he reduceth thy dayes into a short number."; first use referencing a calculating machine in 1897

From the Wikipedia page on Computer A **computer** is a digital electronic machine that can be programmed to carry out L sequences of arithmetic or logical operations (computation) automatically. Modern computers can perform generic sets of L operations known as programs. These programs enable computers to perform a wide range of tasks. A computer system is a I "complete" computer that includes the hardware, operating system (main software), and peripheral equipment needed and used for "full" operation. This term may also refer to a group of computers that are linked and function together, such as a computer L network or computer cluster. A broad range of industrial and consumer products use computers as control systems.

Simple special-purpose devices like

microwave ovens and remote controls are



included, as are factory devices like industrial robots and computer-aided design, as well as general-purpose devices like personal computers and mobile devices like

smartphones. Computers power the Internet, which links billions of other computers and users.

Early computers were meant to be used only for calculations. Simple manual I instruments like the abacus have aided people in doing calculations since ancient times. Early in the Industrial Revolution, some mechanical devices were built to automate long tedious tasks, such as guiding patterns for looms. More sophisticated electrical machines did specialized analog calculations in the early 20th century. The first digital electronic calculating machines were developed during World War II. The first semiconductor transistors in the late 1940s were followed by the silicon-based MOSFET (MOS transistor) and monolithic integrated circuit (IC) chip technologies in the late 1950s, leading to the microprocessor and the microcomputer revolution in the 1970s. The speed, power and versatility of computers have been increasing dramatically ever since then, with transistor counts increasing at a rapid pace (as predicted by Moore's law), leading to the Digital Revolution during the late 20th to early 21st centuries.

Conventionally, a modern computer consists of at least one processing element, typically a central processing unit (CPU) in the form of a microprocessor, along
with some type of computer memory, typically semiconductor memory chips. The Edition 0.35
processing element carries out arithmetic and logical operations, and a sequencing and control unit can change the order of operations in response to stored information. Peripheral devices include input devices (keyboards, mice, joystick, etc.), output devices (monitor screens, printers, etc.), and input/output devices that perform both functions (e.g., the 2000s-era touchscreen). Peripheral devices allow information to be retrieved from an external source and they enable the result of operations to be saved and retrieved.

Text is available under the Creative Commons Attribution-ShareAlike License

concatenate

L

keyword: SQL reserved word (z/OS)

Concurrent CP/M

abbreviation

operating system

Proprietary OS by Digital Research, Inc.

the successor of CP/M-80 and MP/M-80 - Wikipedia: List of operating systems

CP/M

Concurrent CP/M 8-16

abbreviation

operating system

Proprietary OS by Digital Research, Inc.

a dual-processor variant of Concurrent CP/M for 8086 and 8080 CPUs. —<u>Wikipedia: List of</u> operating systems CP/M

Concurrent CP/M-68K

abbreviation

operating system

Proprietary OS by Digital Research, Inc.

a variant for the 68000 - Wikipedia: List of operating systems

CP/M

Concurrent CP/M-86

abbreviation

operating system

Proprietary OS by Digital Research, Inc.

the successor of CP/M-86 and MP/M-86 - Wikipedia: List of operating systems

CP/M

Concurrent DOS

abbreviation

operating system

Proprietary OS by Digital Research, Inc.

the successor of Concurrent CP/M-86 with PC-MODE - Wikipedia: List of operating systems

DOS

Concurrent DOS 8-16

abbreviation

operating system

Proprietary OS by Digital Research, Inc.

a dual-processor variant of Concurrent DOS for 8086 and 8080 CPUs —<u>Wikipedia: List of operating</u> systems

DOS

Concurrent DOS 286

abbreviation

operating system

Proprietary OS by Digital Research, Inc.

Concurrent DOS for the Intel 80286 - Wikipedia: List of operating systems

DOS

Concurrent DOS 386

abbreviation

operating system

Proprietary OS by Digital Research, Inc.

Concurrent DOS for the Intel 80386 — Wikipedia: List of operating systems

DOS

Concurrent DOS 386/MGE

abbreviation

operating system

Proprietary OS by Digital Research, Inc.

a Concurrent DOS 386 variant with advanced graphics terminal capabilities — Wikipedia: List of operating systems

DOS

Concurrent DOS 68K

abbreviation

operating system

Proprietary OS by Digital Research, Inc.

a port of Concurrent DOS to Motorola 68000 CPUs with DOS source code portability capabilities —<u>Wikipedia: List of operating systems</u>

DOS

Concurrent PC-DOS

abbreviation

operating system

Proprietary OS by Digital Research, Inc.

a Concurrent DOS variant for IBM compatible PCs - Wikipedia: List of operating systems

DOS

Concurrent DOS XM

abbreviation

operating system

Proprietary OS by Digital Research, Inc.

a real-mode variant of Concurrent DOS with EEMS support -Wikipedia: List of operating systems

DOS

cond

keyword: SQL reserved word (z/OS)

See: COND — PL/I condition

See: COND — PL/I attribute

CONDITION (name)

keyword: PL/I keyword

Abbreviated: COND

Condition

CONDITION

keyword: PL/I keyword

Abbreviated: COND

Attribute

configuration

keyword: COBOL reserved word (z/OS)

keyword: VHDL reserved key word

confirm

keyword: JavaScript windows method

CONJG(x)

keyword: PL/I keyword

Built-in function

See: CONN — PL/I

connect

keyword: SQL reserved word (ANSI, Oracle)

CONNECTED

keyword: PL/I keyword

Abbreviated: CONN

Attribute

CONSECUTIVE

keyword: PL/I keyword

const

keyword: Java reserved key word keyword: JavaScript reserved word keyword: Objective-C reserved word keyword: PHP reserved word

constant

keyword: Ada reserved key word keyword: VHDL reserved key word

CONTROLLED

keyword: PL/I keyword

Abbreviated: CTL

Attribute

constructor

keyword: JavaScript windows object

contains

keyword: COBOL reserved word (z/OS)

content

keyword: COBOL reserved word (z/OS)

Contiki

operating system Proprietary OS used by Amstrad. Proprietary OS used by Atair for 8-bit, Portfolio. Proprietary OS used by Apple for the Apple II family.

continue

keyword: Java reserved key word

keyword: JavaScript reserved word

keyword: Objective-C reserved word

keyword: PHP reserved word

continuous controller

keyword: MIDI comamnd

B0 hex (where n is the channel number)

2 paraaemters

paramater 1: controller #

paramater 2: controller value

control

keyword: COBOL reserved word (z/OS)

controls

keyword: COBOL reserved word (z/OS)

See: CONV — PL/I

Convergent technologies Operating System

operating system

Proprietary OS by Convergent Technologies.

converting

keyword: COBOL reserved word (z/OS)

CONVERSION

keyword: PL/I keyword

Abbreviated: CONV

Condition

Copland

operating system

Proprietary OS by Apple for the Apple Macintosh.

Apple Macintosh

189

copy

keyword: COBOL reserved word (z/OS)

COPY

keyword: PL/I keyword

Option of GET statement

Coral

keyword: HTML color name



CornflowerBlue

keyword: HTML color name

CornflowerBlue #6495ED

Cornsilk



#FFF8DC

corr

keyword: COBOL reserved word (z/OS)

corresponding

keyword: SQL reserved word (z/OS)

keyword: COBOL reserved word (z/OS)

COS(x)

keyword: PL/I keyword

Built-in function

COSD(x)

keyword: PL/I keyword

Built-in function

COSH(x)

keyword: PL/I keyword

Built-in function

count

Edition 0.35

Milo's Dictionary (with Programming Lexicons)

keyword: COBOL reserved word (z/OS)

keyword: SQL reserved word (z/OS)

COUNT(file-name)

keyword: PL/I keyword

Built-in function

COUNTER

keyword: PL/I keyword

Built-in function

CP-6

operating system

Proprietary OS by Honeywell

CPC Coding scheme

abbreviation

programming language

year created: 1948

created by: Howard H. Aiken

predecessors: Analytical Engine order code

See: CPLN — PL/I

See: CPLX — PL/I

CP/M

abbreviation

operating system

Proprietary OS by Digital Research, Inc.

The original CP/M ran on Intel 8080/8085 and Zilog Z80. -Wikipedia: List of operating systems

Also the name used with an entire family of operating systems.

CP/M 2.2

abbreviation

operating system

Proprietary OS by Amstrad

CP/M

CP/M-68K

abbreviation

operating system

Proprietary OS by Digital Research, Inc.

CP/M for Motorola 68000. - Wikipedia: List of operating systems

CP/M

CP/M-86

Milo's Dictionary (with Programming Lexicons)

abbreviation

operating system

Proprietary OS by Digital Research, Inc.

CP/M-86 CP/M for Intel 8088/8086 — Wikipedia: List of operating systems

CP/M

CP/M-86 Plus

abbreviation

operating system

Proprietary OS by Digital Research, Inc.

CP/M-86 CP/M for Intel 8088/8086 for BDOS 3.0 - Wikipedia: List of operating systems

CP/M

CP/M-8000

abbreviation

operating system

Proprietary OS by Digital Research, Inc.

CP/M-8000 CP/M for Zilog Z8000 - Wikipedia: List of operating systems

CP/M

CP/M Plus

abbreviation

Milo's Dictionary (with Programming Lexicons)

operating system

Proprietary OS by Digital Research, Inc.

Proprietary OS licensed by Amstrad

CP/M for Intel 8080/8085 and Zilog Z80 with BDOS 3.0 - Wikipedia: List of operating systems

CP/M

See: .cpt — file extension

CPU

abbreviation

Central Processing Unit: the central computational and control unit of a computer system, the CPU inteprets and executes the instructions, performs the basic operations of the system, exchanges data with the system's memory and/or peripherals, and manages the system's other components.

Synonym processor

See: .cr2 🛅 — file extension

create

keyword: SQL reserved word (ANSI, Oracle)

See: .crf — file extension

Crimson



#DC143C

Cromemco DOS (CDOS)

operating system

Proprietary OS by Cromemco.

A disk operating system compatible with CP/M. - Wikipedia: List of operating systems

CP/M

Cromix

operating system

Proprietary OS by Cromemco.

A multitasking, mutli-user, Unix-like OS for Cromemco microcomputers with Z80A and/or 68000 CPU. —<u>Wikipedia: List of operating systems</u>

See: .crw 🛅 — file extension

Crypto

keyword: Crypto (initial cap) JavaScript windows object

See: .csh 🖻 — file extension

See: .csl 🕅 — file extension

See: CSTG — PL/I

See: .csv — file extension

See: CTL

CTL360

keyword: PL/I keyword

Option of ENVIRONMENT attribute

CTLASA

keyword: PL/I keyword

Option of ENVIRONMENT attribute

CTOS

abbreviation

operating system

Proprietary OS by Datapoint.

Cassette Tape Operating System for the Datapoint 2200. - Wikipedia: List of operating systems

See: CTSS — operating system

current

keyword: SQL reserved word (ANSI, Oracle)

current_date

keyword: SQL reserved word (z/OS)

CURRENTSTORAGE(variable)

keyword: PL/I keyword

Abbreviated: CSTG

Built-in function

current_time

keyword: SQL reserved word (z/OS)

current_timestamp

keyword: SQL reserved word (z/OS)

currency

keyword: COBOL reserved word (z/OS)

Curry notation system

programming language

year created: 1948

created by: Haskell Curry

predecessors: ENIAC coding system

cursor

keyword: SQL reserved word (z/OS)

Cyan



Dd

D

ASCII: 44 hex

UNICODE: LATIN CAPITAL LETTER D

d

ASCII: 64 hex

UNICODE: LATIN SMALL LETTER D

D

keyword: PL/I keyword

Option of ENVIRONMENT attribute

DarkBlue



DarkCyan

keyword: HTML color name



DarkGoldenRod

keyword: HTML color name



DarkGray

keyword: HTML color name



DarkGrey



DarkGreen

keyword: HTML color name



DarkKhaki

keyword: HTML color name



DarkMagenta

keyword: HTML color name



DarkOliveGreen

keyword: HTML color name

DarkOliveGreen #556B2F

DarkOrange

keyword: HTML color name



DarkOrchid

keyword: HTML color name



DarkRed

keyword: HTML color name



DarkSalmon

keyword: HTML color name

DarkSalmon #E9967A

DarkSeaGreen

Edition 0.35

keyword: HTML color name



DarkSlateBlue

keyword: HTML color name



DarkSlateGray

keyword: HTML color name

DarkSlateGray #2F4F4F

DarkSlateGrey

keyword: HTML color name

DarkSlateGrey #2F4F4F

DarkTurquoise



DarkViolet

keyword: HTML color name



See: .dash — file extension

See: .dat (data) — file extension

See: .dat (video) 📟 — file extension

data

keyword: COBOL reserved word (z/OS)

keyword: SQL reserved word (ANSI, Oracle)

DATA

keyword: PL/I keyword

Stream I/O transmission mode

DATAFIELD

keyword: PL/I keyword

Datapac Multiuser DOS

operating system

Proprietary OS by Digital Research, Inc.

DOS

Datapac System Manager

operating system

Proprietary OS by Digital Research, Inc.

a derivative of Datapac Multiuser DOS - Wikipedia: List of operating systems

DOS

dataxel

keyword: SQL reserved word (z/OS)

date

keyword: JavaScript object

keyword: SQL reserved word (z/OS)

DATE

keyword: PL/I keyword

Built-in function

date-compiled

keyword: COBOL reserved word (z/OS)

date-written

keyword: COBOL reserved word (z/OS)

day

keyword: COBOL reserved word (z/OS)

keyword: SQL reserved word (z/OS)

day-of-week

keyword: COBOL reserved word (z/OS)

DB

keyword: PL/I keyword

Option of ENVIRONMENT attribute

See: .db — file extension

See: .dbf — file extension

dbcs 207

keyword: COBOL reserved word (z/OS)

dbms_in_memory

keyword: SQL reserved word (z/OS)

See: DCL — PL/I

See: .dcr 🛅 — file extension

See: .dds 🚰 — file extension

de

keyword: COBOL reserved word (z/OS)

See: .deb — file extension

debug-contents

keyword: COBOL reserved word (z/OS)

debugger

keyword: JavaScript reserved word

debug-item

keyword: COBOL reserved word (z/OS)

debug-line

keyword: COBOL reserved word (z/OS)

debug-name

keyword: COBOL reserved word (z/OS)

debug-sub-1

keyword: COBOL reserved word (z/OS)

debug-sub-2

keyword: COBOL reserved word (z/OS)

debug-sub-3

keyword: COBOL reserved word (z/OS)

debugging

keyword: COBOL reserved word (z/OS)

dec

keyword: SQL reserved word (z/OS)

See: DEC — PL/I attribute

See: DEC — PL/I function

decimal

keyword: SQL reserved word (ANSI, Oracle, z/OS)

DECIMAL

keyword: PL/I keyword

Abbreviated: DEC

Attribute

DECIMAL(a[,p[,q]])

keyword: PL/I keyword

Abbreviated: DEC

Built-in function

decimal-point

keyword: COBOL reserved word (z/OS)

declaratives

keyword: COBOL reserved word (z/OS)

declare

keyword: Ada reserved key word

keyword: PHP reserved word

keyword: SQL reserved word (z/OS)

DECLARE

keyword: PL/I keyword

Abbreviated: DCL

Statement

DEC MICA

abbreviation

operating system

Proprietary OS by Digital Equipment Corporation (DEC). Later owned by Compaq, Hewlett-Packard, and Hewlett Packard Enterprise

for the DEC PRISM - Wikipedia: List of operating systems

decodeURI

keyword: JavaScript windows method

decodeURIComponent

keyword: JavaScript windows method

DeepPink

keyword: HTML color name



DeepSkyBlue

keyword: HTML color name



def

keyword: Ruby reserved word

See: DEF — PL/I

default

keyword: Java reserved key word

keyword: JavaScript reserved word

keyword: Objective-C reserved word

keyword: PHP reserved word

keyword: SQL reserved word (ANSI, Oracle, z/OS)

DEFAULT

keyword: PL/I keyword

Abbreviated: DFT

Statement

defaultStatus

keyword: JavaScript windows method

DEFINED

keyword: PL/I keyword

Abbreviated: DEF

Attribute

defined?

keyword: Ruby reserved word

delay

keyword: Ada reserved key word

DELAY (n)

keyword: PL/I keyword

Statement

delete

keyword: COBOL reserved word (z/OS)

keyword: JavaScript reserved word

keyword: SQL reserved word (ANSI, Oracle, z/OS)

DELETE

keyword: PL/I keyword

Statement

delimited

keyword: COBOL reserved word (z/OS)

delimiter

keyword: COBOL reserved word (z/OS)

delta

keyword: Ada reserved key word

Deos

operating system

Proprietary OS by DDC-I, Inc.

Time & Space Partitioned RTOS, Certified to DO-178B, Level A since 1998. - Wikipedia: List of

depending

keyword: COBOL reserved word (z/OS)

DEPTH

keyword: PL/C keyword

Diagnostic option of PUT statement as in OUT SNAP DEPTH (4);

desc

keyword: SQL reserved word (ANSI, Oracle, z/OS)

descending

keyword: COBOL reserved word (z/OS)

DESCRIPTION

keyword: PL/I keyword

Option of DEFAULT statement

destination

keyword: COBOL reserved word (z/OS)

detail

keyword: COBOL reserved word (z/OS)

See: DFT — PL/I

DG/UX

abbreviation

operating system

Proprietary OS by Data General.

See: .dib 🔽 — file extension

die()

keyword: PHP reserved word (this is a reserved word and not a function)

Digital UNIX

operating system

Proprietary OS by Digital Equipment Corporation (DEC). Later owned by Compaq, Hewlett-Packard, and Hewlett Packard Enterprise

derived from OSF/1, became HP's Tru64 UNIX - Wikipedia: List of operating systems

digits

keyword: Ada reserved key word
DIM(x, n)

keyword: PL/I keyword

Built-in function

DimGray

keyword: HTML color name



DimGrey

keyword: HTML color name



See: _DIR_ — PHP

DIRECT

keyword: PL/I keyword

Attribute

directive_error

disconnect

keyword: VHDL reserved key word

display

keyword: COBOL reserved word (z/OS)

DISPLAY

keyword: PL/I keyword

Statement

display-1

keyword: COBOL reserved word (z/OS)

distinct

keyword: SQL reserved word (ANSI, Oracle, z/OS)

divide

keyword: COBOL reserved word (z/OS)

DIVIDE(**x**, **y**, **p**[,q])

keyword: PL/I keyword

Built-in function

division

keyword: COBOL reserved word (z/OS)

See: .djvu 🂐 — file extension

See: .dm — file extension

See: .dng — file extension

do

keyword: Ada reserved key word

keyword: Java reserved key word

keyword: JavaScript reserved word

keyword: Objective-C reserved word

keyword: PHP reserved word

keyword: PL/I keyword

Statement

keyword: Ruby reserved word

keyword: SQL reserved word (z/OS)

Document

keyword: Document (initial cap) JavaScript windows object

219

DodgerBlue

keyword: HTML color name



DOM

keyword: JavaScript windows object

Domain/OS

operating system

Proprietary operating system by Apollo Computer.

One of the first network-based systems. Ran on Apollo/Domain hardware. Later bought by Hewlett-Packard. —<u>Wikipedia: List of operating systems</u>

DOS

abbreviation

operating system

Proprietary OS by Digital Research, Inc.

A family of operating systems.

DOS

abbreviation

```
operating system
```

Proprietary OS by Datapoint.

Disk Operating System for the Datapoint 2200, 5500, and 1100. —<u>Wikipedia: List of operating</u> systems

DOS PLus 1.1-2.1

abbreviation

operating system

Proprietary OS by Digital Research, Inc.

a single-user, multi-tasking system derived from Concurrent DOS 4.1 Đ 5.0 — Wikipedia: List of operating systems

DOS

double

keyword: Java reserved key word

keyword: JavaScript reserved word

keyword: Objective-C reserved word

keyword: SQL reserved word (z/OS)

doublegreatethan

keyword: SQL reserved word (z/OS)

down

downto

keyword: VHDL reserved key word

DR-DOS 3.31-6.0

abbreviation

operating system

Proprietary OS by Digital Research, Inc.

a single-user, single-tasking native DOS derived from Concurrent DOS 6.0 — <u>Wikipedia: List of</u> operating systems

DOS

drop

keyword: SQL reserved word (ANSI, Oracle)

Drum SCOPE

operating system

Proprietary OS by Control Data Corporation.

Ran on the Upper 3000 series. - Wikipedia: List of operating systems

SCOPE

See: .drw 🕅 — file extension

duplicates

keyword: COBOL reserved word (z/OS)

See: .dvr 🌌 — file extension

See: .dvf — file extension

dynamic

keyword: COBOL reserved word (z/OS)

See: @dynamic — Objective C



E

ASCII: 45 hex

UNICODE: LATIN CAPITAL LETTER E

e

ASCII: 65 hex

UNICODE: LATIN SMALL LETTER E

E (w, d[s])

keyword: PL/I keyword

Format item

echo

keyword: PHP reserved word

EDIT 224

Stream I/O transmission mode

EDSAC Initial Orders

abbreviation

programming language

year created: 1949

created by: David Wheeler

predecessors: ENIAC coding system

egcs

keyword: COBOL reserved word (z/OS)

egi

keyword: COBOL reserved word (z/OS)

eject

keyword: COBOL reserved word (z/OS)

Element

keyword: Element (initial cap) JavaScript windows object

else

keyword: Ada reserved key word

keyword: COBOL reserved word (z/OS)

keyword: Java reserved key word

keyword: JavaScript reserved word

keyword: Objective-C reserved word

keyword: PHP reserved word

keyword: PL/I keyword

Clause of IF statement

keyword: Ruby reserved word

keyword: SQL reserved word (ANSI, Oracle, z/OS)

keyword: VHDL reserved key word

elseif

keyword: PHP reserved word

keyword: SQL reserved word (z/OS)

elsif

keyword: Ada reserved key word keyword: Ruby reserved word

S

keyword: VHDL reserved key word

See: .email — file extension

Embed

keyword: Embed (initial cap) JavaScript windows object

embeds

keyword: JavaScript windows property

See: .emf 🚨 — file extension

EMPTY

keyword: PL/I keyword

Built-in function

emi

keyword: COBOL reserved word (z/OS)

See: .eml — file extension

See: .emix — file extension

empty()

keyword: PHP reserved word (this is a reserved word and not a function)

See: .emz — file extension

enable

keyword: COBOL reserved word (z/OS)

encodeURI

keyword: JavaScript windows method

encodeURIComponent

keyword: JavaScript windows method

See: _ENCODING_ — Ruby

end

keyword: Ada reserved key word

keyword: COBOL reserved word (z/OS)

keyword: PL/I keyword

Statement

keyword: END (all caps) Ruby reserved word

keyword: end (all lower case) Ruby reserved word

keyword: SQL reserved word (z/OS)

keyword: VHDL reserved key word

See: @end — Objective C

keyword: COBOL reserved word (z/OS)

end-call

keyword: COBOL reserved word (z/OS)

end-compute

keyword: COBOL reserved word (z/OS)

enddeclare

keyword: PHP reserved word

end-delete

keyword: COBOL reserved word (z/OS)

end-divide

keyword: COBOL reserved word (z/OS)

end-evaluate

keyword: COBOL reserved word (z/OS)

ENDFILE (file expression)

keyword: PL/I keyword

Condition

endfor

keyword: PHP reserved word

endforeach

keyword: PHP reserved word

endif

keyword: PHP reserved word

end-if

keyword: COBOL reserved word (z/OS)

end-invoke

keyword: COBOL reserved word (z/OS)

end-multiply

endofdata

keyword: SQL reserved word (z/OS)

end-of-page

keyword: COBOL reserved word (z/OS)

ENDPAGE (file expression)

keyword: PL/I keyword

Condition

end-perform

keyword: COBOL reserved word (z/OS)

end-read

keyword: COBOL reserved word (z/OS)

end-receive

keyword: COBOL reserved word (z/OS)

end-return

end-rewrite

keyword: COBOL reserved word (z/OS)

end-search

keyword: COBOL reserved word (z/OS)

end-start

keyword: COBOL reserved word (z/OS)

end-string

keyword: COBOL reserved word (z/OS)

end-subtract

keyword: COBOL reserved word (z/OS)

endswitch

keyword: PHP reserved word

end-unstring

end-write

keyword: COBOL reserved word (z/OS)

ending

keyword: COBOL reserved word (z/OS)

ENIAC coding system

abbreviation

programming language

year created: 1943-1946

created by: John von Neumann, John Mauchly, J. Presper Eckert, Herman Goldstine after Alan Turing. The first programmers of ENIAC were Kay McNulty, Betty Jennings, Betty Snyder, Marlyn Meltzer, Fran Bilas, and Ruth Lichterman.

predecessors: none (unique language)

ENIAC Short Code

abbreviation

programming language

year created: 1946

created by: Richard Clippinger, John von Neumann after Alan Turing

predecessors: ENIAC coding system

ensure

keyword: Ruby reserved word

enter

keyword: COBOL reserved word (z/OS)

entity

keyword: VHDL reserved key word

entry

keyword: Ada reserved key word keyword: COBOL reserved word (z/OS) keyword: PL/I keyword Attribute, statement

enum

keyword: Java reserved key word

keyword: JavaScript reserved word

keyword: Objective-C reserved word

environment

keyword: PL/I keyword

keyword: COBOL reserved word (z/OS)

Abbreviate: ENV

Attribute

See: .eps 🔟 — file extension

eof

keyword: SQL reserved word (z/OS)

eop

keyword: COBOL reserved word (z/OS)

equal

keyword: COBOL reserved word (z/OS)

ERF(x)

keyword: PL/I keyword

Built-in function

ERFC(x)

keyword: PL/I keyword

Built-in function

error

keyword: COBOL reserved word (z/OS)

keyword: PL/I keyword

escape

keyword: JavaScript depreciated function

keyword: SQL reserved word (z/OS)

esi

keyword: COBOL reserved word (z/OS)

eval

keyword: JavaScript reserved word

eval()

keyword: PHP reserved word (this is a reserved word and not a function)

evaluate

keyword: COBOL reserved word (z/OS)

Event

keyword: Event (all caps) JavaScript windows Object

EVENT

236

keyword: PL/I keyword

Attribute, option of CALL, DELETE, DISPLAY, READ, REWRITE, WRITE statements

every

keyword: COBOL reserved word (z/OS)

except

keyword: SQL reserved word (z/OS)

exception

keyword: Ada reserved key word

keyword: COBOL reserved word (z/OS)

exclusive

keyword: SQL reserved word (Oracle)

keyword: PL/I keyword

Attribute

See: .exe — file extension

exists

keyword: SQL reserved word (Oracle)

exit

keyword: Ada reserved key word keyword: COBOL reserved word (z/OS) keyword: PL/I keyword Statement keyword: VHDL reserved key word

exit()

keyword: PHP reserved word (this is a reserved word and not a function)

EXP(x)

keyword: PL/I keyword

Built-in function

export

keyword: JavaScript reserved word

See: EXT — PL/I

extend

keyword: COBOL reserved word (z/OS)

extends

238

keyword: Java reserved key word

keyword: JavaScript reserved word

keyword: PHP reserved word

EXTENTNUMBER (n)

keyword: PL/I keyword

Option of ENVIRONMENT attribute

extern

keyword: Objective-C reserved word

external

keyword: COBOL reserved word (z/OS)

keyword: PL/I keyword

Abbreviated: EXT

Attribute



F

ASCII: 46 hex

UNICODE: LATIN CAPITAL LETTER F

f

ASCII: 66 hex

UNICODE: LATIN SMALL LETTER F

\mathbf{F}

keyword: PL/I keyword

Option of ENVIRONMENT attribute

F (x[,y[,z]]))

keyword: PL/I keyword

Format item

false

keyword: COBOL reserved word (z/OS) keyword: Java reserved literal value word keyword: JavaScript reserved word keyword: Ruby reserved word keyword: SQL reserved word (z/OS)

FB

keyword: PL/I keyword Option of ENVIRONMENT attribute

FBS

keyword: PL/I keyword

Option of ENVIRONMENT attribute

fd

keyword: COBOL reserved word (z/OS)

file

keyword: COBOL reserved word (z/OS)

keyword: SQL reserved word (Oracle)

keyword: VHDL reserved key word

See: _FILE_ — PHP and Ruby

file-control

keyword: COBOL reserved word (z/OS)

fileUpload

keyword: JavaScript windows method

filler

keyword: COBOL reserved word (z/OS)

final

keyword: COBOL reserved word (z/OS)

keyword: Java reserved key word

keyword: JavaScript reserved word

keyword: PHP reserved word

finally

keyword: Java reserved key word

keyword: JavaScript reserved word

keyword: PHP reserved word

See: @finally — Objective C

Fire OS

operating system

Proprietary OS by Amazon

FireBrick

keyword: HTML color name



first

keyword: COBOL reserved word (z/OS)

keyword: SQL reserved word (z/OS)

See: .fla 🗾 — file extension

See: .flac

FlexOS

operating system

Proprietary OS by Digital Research, Inc.

1.0 D 2.34, a derivative of Concurrent DOS 286 - Wikipedia: List of operating systems

DOS

FlexOS 186

operating system

Proprietary OS by Digital Research, Inc.

a variant of FlexOS for terminals - Wikipedia: List of operating systems

DOS

FlexOS 286

operating system

Proprietary OS by Digital Research, Inc.

a variant of FlexOS for hosts -Wikipedia: List of operating systems

DOS

FlexOS 386

operating system

Proprietary OS by Digital Research, Inc.

a later variant of FlexOS for hosts - Wikipedia: List of operating systems

DOS

FlexOS 68K

operating system

Proprietary OS by Digital Research, Inc.

a derivative of Concurrent DOS 68K - Wikipedia: List of operating systems

DOS

float

- keyword: Java reserved key word
- keyword: JavaScript reserved word
- keyword: Objective-C reserved word
- keyword: SQL reserved word (Oracle)

FloralWhite

keyword: HTML color name



FLOW

keyword: PL/C keyword

Condition as in ON FLOW BEGIN;

Diagnostic option of PUT statement as in PUT FLOW;

Statement as in FLOW (15, 20);

See: .flv 💋

— file extension

See: FMS — operating system

focus

keyword: JavaScript windows method

footing

keyword: COBOL reserved word (z/OS)

for

- keyword: Ada reserved key word
- keyword: COBOL reserved word (z/OS)
- keyword: Java reserved key word
- keyword: JavaScript reserved word
- keyword: Objective-C reserved word
- keyword: PHP reserved word
- keyword: Ruby reserved word
- keyword: SQL reserved word (ANSI, Oracle, z/OS)
- keyword: VHDL reserved key word

foreach

keyword: PHP reserved word

ForestGreen

keyword: HTML color name 246



forms

keyword: JavaScript windows read-only property

FORTRAN Monitor System (FMS)

operating system

Proprietary OS by IBM

for the IBM 709 and 7090 - Wikipedia: List of operating systems

See: .fpx 🗟 — file extension

Frame

keyword: Frame (initial cap) JavaScript windows object

frames

keyword: JavaScript windows property

frameRate

keyword: JavaScript windows method

from

keyword: COBOL reserved word (z/OS)

keyword: SQL reserved word (ANSI, Oracle, z/OS)

Fuchsia

keyword: HTML color name



full

keyword: SQL reserved word (z/OS)

function

keyword: Ada reserved key word

keyword: COBOL reserved word (z/OS)

keyword: JavaScript reserved word

keyword: PHP reserved word

keyword: VHDL reserved key word

See: _FUNCTION_ — PHP

Fuchsia

operating system

Proprietary OS by Google

Fuchsia is a capability-based, real-time, operating system (RTOS) scalable to universal devices, in early development, from the tiniest embedded hardware, wristwatches, tablets to the largest personal computers. Unlike Chrome OS and Android, it is not based on the Linux kernel, but instead began on a new microkernel called "Zircon", derived from "Little Kernel". —<u>Wikipedia: List of</u> operating systems

Gg

G

ASCII: 47 hex

UNICODE: LATIN CAPITAL LETTER G

g

ASCII: 67 hex

UNICODE: LATIN SMALL LETTER G

Gainsboro

keyword: HTML color name

Gainsboro #DCDCDC

gate

 $g\acute{\epsilon}jt$ (Received Pronunciation and General American), *countable noun*

1. logic An idealized model of computation implementing a Boolean or mathematical function.

2. *electronics* A physical electronic switch or device that allows or prevents the flow of current in a circuit when specific input conditions are met.

3. *electronics* An electrode in a semiconductor, especially a field-effect transistor, that modulates or controls the current flowing through the transistor according to the voltage applied to the electrode.

4. *electronics* A signal that makes an electronic circuit operative or inoperative either for a certain time interval or until another signal is received.

verb

1. electronics To control the operation of an electronic device by means of a gate.

2. *electronics* To select the parts of a wave signal that are within a certain range of amplitude, frequency, or within certain time intervals.

generate

keyword: COBOL reserved word (z/OS)

keyword: VHDL reserved key word

GEOS

abbreviation

operating system

Proprietary OS by Commodore International.

generic

keyword: Ada reserved key word

keyword: VHDL reserved key word

GhostWhite

keyword: HTML color name



giving

keyword: COBOL reserved word (z/OS)

gLinux

operating system

Proprietary OS by Google

a Linux distribution that Google uses internally - Wikipedia: List of operating systems

global

keyword: COBOL reserved word (z/OS)

keyword: PHP reserved word

GM-NAA I/O

abbreviation

operating system
Proprietary OS by IBM

for the IBM 704 - Wikipedia: List of operating systems

GM OS

abbreviation

operating system

Proprietary OS by IBM

for the IBM 704 - Wikipedia: List of operating systems

GNO/OS

abbreviation

operating system

Proprietary OS by Apple for the Apple II family

g0

keyword: COBOL reserved word (z/OS)

goback

keyword: COBOL reserved word (z/OS)

Gold



GoldenRod

keyword: HTML color name



goto

keyword: Ada reserved key word

keyword: Java reserved key word

keyword: JavaScript reserved word

keyword: Objective-C reserved word

keyword: PHP reserved word

See: .gp — file extension

See: .gp3 — file extension

See: .gp4 🜌 — file extension

See: .gp5 🛃 — file extension

See: .gpt — file extension

See: .gpx — file extension

grant

keyword: SQL reserved word (ANSI, Oracle)

Gray

keyword: HTML color name



greateeq

keyword: SQL reserved word (z/OS)

greater

keyword: COBOL reserved word (z/OS)

Grey

keyword: HTML color name



Green



GreenYellow

keyword: HTML color name



group

keyword: COBOL reserved word (z/OS)

keyword: SQL reserved word (ANSI, Oracle, z/OS)

keyword: VHDL reserved key word

group_by

keyword: SQL reserved word (z/OS)

GS/OS

abbreviation

operating system

Proprietary OS by Apple for the Apple II family

guarded

keyword: VHDL reserved key word

Hh

H

ASCII: 48 hex

UNICODE: LATIN CAPITAL LETTER H

h

ASCII: 68 hex

UNICODE: LATIN SMALL LETTER H

See: .h264 🥝 — file extension

See: __halt_compiler__ - PHP

Harmony OS

operating system

Proprietary OS by Huawei

Harris UNIX

operating system

Proprietary OS by Harris Corporation

Proprietary UNIX based OS for Harris' Computers (MCX)

hash

keyword: SQL reserved word (z/OS)

hasOwnProperty

keyword: JavaScript method

having

keyword: SQL reserved word (ANSI, Oracle, z/OS)

heading

keyword: COBOL reserved word (z/OS)

HeartOS

operating system

Proprietary OS by DDC-I, Inc.

POSIX-based Hard Real-TIme Operating System. - Wikipedia: List of operating systems

HDOS

abbreviation

operating system

Proprietary OS by Heathkit and Zeniith Data Systems

ran on the H8 and Heath/Zenith Z-89 series - Wikipedia: List of operating systems

See: .heic — file extension

See: .hex — file extension

hidden

keyword: JavaScript windows property

high-value

keyword: COBOL reserved word (z/OS)

high-values

keyword: COBOL reserved word (z/OS)

high_values

keyword: SQL reserved word (z/OS)

hintend

keyword: SQL reserved word (z/OS)

hint_error

keyword: SQL reserved word (z/OS)

hintstart

keyword: SQL reserved word (z/OS)

History

keyword: History (initial cap) JavaScript windows object

HoneyDew

keyword: HTML color name



HotPink

keyword: HTML color name



hour

```
keyword: SQL reserved word (z/OS) 261
```

HP Multi-Programming Executive (MPE, MPE/XL, and MPE/iX)

operating system

Proprietary OS by Hewlett-Packard and Hewlett Packard Enterprise

runs on HP 3000 and HP e3000 mini-computers -Wikipedia: List of operating systems

HP-UX

abbreviation

operating system

Proprietary OS by Hewlett-Packard and Hewlett Packard Enterprise

runs on HP9000 and Itanium servers (from small to mainframe-class computers) —<u>Wikipedia: List</u> of operating systems

HT-11

operating system

Proprietary OS by Heathkit and Zeniith Data Systems

a modified version of RT-11 that ran on the Heathkit H11 --- Wikipedia: List of operating systems

See: .htm — file extension

See: .html — file extension

lí

I

ASCII: 49 hex

UNICODE: LATIN CAPITAL LETTER I

i

ASCII: 69 hex

UNICODE: LATIN SMALL LETTER i

IBM 4680 OS

abbreviation

operating system

Proprietary OS by Digital Research, Inc.

a Point of Sale (POS) operating system based on FlexOS - Wikipedia: List of operating systems

DOS

IBM 4690 OS

abbreviation

operating system

Proprietary OS by Digital Research, Inc.

a Point of Sale (POS) operating system based on FlexOS - Wikipedia: List of operating systems

DOS

IBSYS

abbreviation

operating system

Proprietary OS by IBM



id

keyword: COBOL reserved word (z/OS)

keyword: Objective-C word

identification

keyword: COBOL reserved word (z/OS)

identified

keyword: SQL reserved word (Oracle)

if

- keyword: Ada reserved key word
- keyword: COBOL reserved word (z/OS)
- keyword: Java reserved key word
- keyword: JavaScript reserved word
- keyword: Objective-C reserved word
- keyword: PHP reserved word
- keyword: Ruby reserved word
- keyword: SQL reserved word (z/OS)
- keyword: VHDL reserved key word

ifelse

keyword: SQL reserved word (z/OS)

IJMON

abbreviation

operating system

Proprietary OS by IBM

A bootable serial I/O monitor for loading programs for the IBM 1400 series - Wikipedia: List of

operating systems

Image

keyword: Image (initial cap) JavaScript windows object

See: _imaginary — Objective C

immediate

keyword: SQL reserved word (ANSI, Oracle)

IMP

keyword: Objective-C word

implement

keyword: Java reserved key word

implements

keyword: JavaScript reserved word

keyword: PHP reserved word

See: @implementation — Objective C

import

imports

keyword: Java reserved key word

impure

keyword: VHDL reserved key word

IMS Multiuser DOS

abbreviation

operating system

Proprietary OS by Digital Research, Inc.

DOS

IMS REAL/32

operating system

Proprietary OS by Digital Research, Inc.

a derivative of Multiuser DOS - Wikipedia: List of operating systems

DOS

IMS REAL/NG

abbreviation

operating system

Proprietary OS by Digital Research, Inc.

the successor of REAL/32 - Wikipedia: List of operating systems

DOS

in

keyword: Ada reserved key word

keyword: COBOL reserved word (z/OS)

keyword: JavaScript reserved word

keyword: Objective-C former reserved word

keyword: Ruby reserved word

keyword: SQL reserved word (ANSI, Oracle, z/OS)

keyword: VHDL reserved key word

include

keyword: PHP reserved word

include_once

keyword: PHP reserved word

increment

keyword: SQL reserved word (Oracle)

index

keyword: COBOL reserved word (z/OS)

keyword: SQL reserved word (Oracle)

indexed

keyword: COBOL reserved word (z/OS)

IndianRed

keyword: HTML color name



indicate

keyword: COBOL reserved word (z/OS)

Indigo



inertial

keyword: VHDL reserved key word

Inferno

operating system

Proprietary OS by Bell Labs.

Infinity

keyword: JavaScript property

inherits

keyword: COBOL reserved word (z/OS)

initial

keyword: COBOL reserved word (z/OS)

keyword: SQL reserved word (Oracle)

initialize

keyword: COBOL reserved word (z/OS)

initiate

keyword: COBOL reserved word

inline

keyword: Objective-C reserved word

in_memory

keyword: SQL reserved word (z/OS)

inner

keyword: SQL reserved word (z/OS)

innerHeight

keyword: JavaScript windows method

innerWidth

keyword: JavaScript windows method

inout

keyword: Objective-C former reserved word

keyword: VHDL reserved key word

input

keyword: COBOL reserved word (z/OS)

input-output

keyword: COBOL reserved word (z/OS)

insert

keyword: COBOL reserved word (z/OS)

keyword: SQL reserved word (ANSI, Oracle, z/OS)

inspect

keyword: COBOL reserved word (z/OS)

installation

keyword: COBOL reserved word (z/OS)

instanceof

keyword: Java reserved key word

keyword: JavaScript reserved word

keyword: PHP reserved word

insteadof

int

- keyword: Java reserved key word
- keyword: JavaScript reserved word
- keyword: Objective-C reserved word
- keyword: SQL reserved word (z/OS)

integer

keyword: SQL reserved word (ANSI, Oracle, z/OS)

INTEGRITY

operating system

Proprietary OS by Green Hills Software

Reliable Operating system - Wikipedia: List of operating systems

INTEGRITY-178B

operating system

Proprietary OS by Green Hills Software

A DO-178B certified version of INTEGRITY. - Wikipedia: List of operating systems

interface

keyword: Ada reserved key word

keyword: Java reserved key word

keyword: JavaScript reserved word

keyword: PHP reserved word

See: @interface — Objective C

internet

in '-tər-net, noun

A global interconnected computer network using standardized communication protocols.

intersect

keyword: SQL reserved word (ANSI, Oracle, z/OS)

interval

keyword: SQL reserved word (z/OS)

into

keyword: COBOL reserved word (z/OS)

keyword: SQL reserved word (ANSI, Oracle, z/OS)

invalid

274

Edition 0.35

keyword: COBOL reserved word

invoke

keyword: COBOL reserved word (z/OS)

i-0

keyword: COBOL reserved word (z/OS)

i-o-control

keyword: COBOL reserved word (z/OS)

iOS

abbreviation

operating system

Proprietary OS by Apple for iPhone and iPod Touch.

iPad OS

operating system

Proprietary OS by Apple for iPad.

iPhone OS

operating system

Proprietary OS by Apple for iPhone.

iPod (unnamed)

operating system

Unnamed proprietary embedded operating system by Apple for the iPod. —<u>Wikipedia: List of</u> operating systems

iRMX

abbreviation

operating system

Proprietary OS by Intel Corporation

real-time operating system originally created to support the Intel 8080 and 8086 processor families in embedded applications. —<u>Wikipedia: List of operating systems</u>

is

keyword: Ada reserved key word

keyword: COBOL reserved word (z/OS)

keyword: SQL reserved word (ANSI, Oracle, z/OS)

keyword: VHDL reserved key word

isFinite

keyword: JavaScript method

ISIS and ISIS-II

abbreviation

operating system

Proprietary OS by Intel Corporation

"Intel Systems Implementation Supervisor" was an environment for development of software within the Intel microprocessor family in the early 1980s on their Intellec Microcomputer Development System and clones. ISIS-II worked with 8 inch floppy disks and had an editor, cross-assemblers, a linker, an object locator, debugger, compilers for PL/M, a BASIC interpreter, etc. and allowed file management through a console. —<u>Wikipedia: List of operating systems</u>

isNaN

keyword: JavaScript method

See: .iso — file extension

isPrototypeOf

keyword: JavaScript method

isset()

keyword: PHP reserved word (this is a reserved word and not a function)

iterate

keyword: SQL reserved word (z/OS)

Ivory



Jj

J

ASCII: 4A hex

UNICODE: LATIN CAPITAL LETTER J

j

ASCII: 6A hex

UNICODE: LATIN SMALL LETTER J

Jacquard machine

programming language

year created: 1804

created by: Joseph Marie Jacquard

predecessors: none (unique language)

join

keyword: SQL reserved word (z/OS)

See: .jpeg 🖤 v file extension

See: .jpg 🖻 — file extension

just

keyword: COBOL reserved word (z/OS)

justified

keyword: COBOL reserved word (z/OS)

Kk

K

ASCII: 4B hex

UNICODE: LATIN CAPITAL LETTER K

k

ASCII: 6B hex

UNICODE: LATIN SMALL LETTER K

kanji

keyword: COBOL reserved word (z/OS)

key

keyword: COBOL reserved word (z/OS)

Khaki

Khaki #F0E68C

Kronos (Kronographic OS)

operating system

Proprietary OS by Control Data Corporation.

L

ASCII: 4C hex

UNICODE: LATIN CAPITAL LETTER L

1

ASCII: 6C hex UNICODE: LATIN SMALL LETTER L

label

keyword: COBOL reserved word (z/OS)

keyword: VHDL reserved key word

last

keyword: COBOL reserved word (z/OS)

keyword: SQL reserved word (z/OS)

Lavender

keyword: HTML color name



LavenderBlush

keyword: HTML color name

LavenderBlush #FFF0F5

LawnGreen

keyword: HTML color name



leading

keyword: COBOL reserved word (z/OS)

keyword: SQL reserved word (z/OS)

leave

```
keyword: SQL reserved word (z/OS) 284
```

left

keyword: COBOL reserved word (z/OS)

keyword: SQL reserved word (z/OS)

LemonChiffon

keyword: HTML color name

LemonChiffon #FFFACD

length

keyword: COBOL reserved word (z/OS)

keyword: JavaScript property

less

keyword: COBOL reserved word (z/OS)

lesseq

keyword: SQL reserved word (z/OS)

let

level

keyword: SQL reserved word (ANSI, Oracle)

See: .lib — file extension

library

keyword: VHDL reserved key word

LightBlue

keyword: HTML color name



LightCoral

keyword: HTML color name



LightCyan



LightGoldenRodYellow

keyword: HTML color name

LightGoldenRodYellow #FAFAD2

LightGray

keyword: HTML color name



LightGrey

keyword: HTML color name



LightGreen



LightPink

keyword: HTML color name



LightSalmon

keyword: HTML color name



LightSeaGreen

keyword: HTML color name

LightSeaGreen #20B2AA

LightSkyBlue


keyword: HTML color name



LightSlateGrey

keyword: HTML color name

LightSlateGrey #778899

LightSteelBlue

keyword: HTML color name

LightSteelBlue #B0C4DE

LightYellow

keyword: HTML color name



like

keyword: SQL reserved word (ANSI, Oracle, z/OS)

Lime

keyword: HTML color name



LimeGreen

keyword: HTML color name



limit

keyword: COBOL reserved word (z/OS)

limited

keyword: Ada reserved key word

limits

keyword: COBOL reserved word (z/OS)

linage

keyword: COBOL reserved word (z/OS)

linage-counter

keyword: COBOL reserved word (z/OS)

line

keyword: COBOL reserved word (z/OS)

See: _LINE_ — PHP and Ruby

line-counter

keyword: COBOL reserved word (z/OS)

Linen

keyword: HTML color name



lines

link

keyword: JavaScript windows non-standard method

linkage

keyword: COBOL reserved word (z/OS)

keyword: VHDL reserved key word

list()

keyword: PHP reserved word (this is a reserved word and not a function)

literal

keyword: VHDL reserved key word

Lite OS

operating system

Proprietary OS by Huawei

local-storage

keyword: COBOL reserved word (z/OS)

location

keyword: JavaScript windows object

lock

keyword: COBOL reserved word (z/OS)

keyword: SQL reserved word (Oracle)

See: .logic 🗟 — file extension

long

keyword: Java reserved key word

keyword: JavaScript reserved word

keyword: Objective-C reserved word

keyword: SQL reserved word (Oracle)

loop

keyword: Ada reserved key word

S

keyword: SQL reserved word (z/OS)

keyword: VHDL reserved key word

lower

293

keyword:

lower_mb

keyword: SQL reserved word (z/OS)

low-value

keyword: COBOL reserved word (z/OS)

low-values

keyword: COBOL reserved word (z/OS)

low_values

keyword: SQL reserved word (z/OS)

lpad

keyword: SQL reserved word (z/OS)

See: .lst — file extension

Mm

Μ

ASCII: 4D hex

UNICODE: LATIN CAPITAL LETTER M

m

ASCII: 6D hex

UNICODE: LATIN SMALL LETTER M



operating system

Proprietary OS by Control Data Corporation.

macOS

operating system

Proprietary OS by Apple for the Apple Macintosh.

Apple Macintosh

macOS Server

operating system

Proprietary OS by Apple for the Apple Macintosh.

Apple Macintosh

Mac OS X

operating system

Proprietary OS by Apple for the Apple Macintosh.

Apple Macintosh

Mac OS X Server

operating system

Apple Macintosh

Magenta

keyword: HTML color name



magnusoft ZETA

operating system

Proprietary OS devloped by yellowTAB, based on BeOS r5.1d0 source code. —<u>Wikipedia: List of</u> operating systems

See:.map — file extension

map

keyword: VHDL reserved key word

Maroon

keyword: HTML color name



Math

keyword: JavaScript object

max

keyword: SQL reserved word (z/OS)

maxextents

keyword: SQL reserved word (Oracle)

See: .mdb — file extension

See: .mdi 🗟 — file extension

MediumAquaMarine

keyword: HTML color name



MediumBlue

keyword: HTML color name



keyword: HTML color name



MediumPurple

keyword: HTML color name

MediumPurple #9370DB

MediumSeaGreen

keyword: HTML color name

MediumSeaGreen #3CB371

MediumSlateBlue

keyword: HTML color name

MediumSlateBlue #7B68EE

MediumSpringGreen

keyword: HTML color name



MediumTurquoise

keyword: HTML color name

MediumTurquoise #48D1CC

MediumVioletRed

keyword: HTML color name

MediumVioletRed #C71585

memory

keyword: COBOL reserved word (z/OS)

memory_size

keyword: SQL reserved word (z/OS)

merge

keyword: COBOL reserved word (z/OS)

keyword: SQL reserved word (z/OS)

message

keyword: COBOL reserved word (z/OS)

metaclass

keyword: COBOL reserved word (z/OS)

meta_hint

keyword: SQL reserved word (z/OS)

method

keyword: COBOL reserved word (z/OS)

See: _METHOD_ — PHP

method-id

keyword: COBOL reserved word (z/OS)

See: .mid — file extension

See: .midi — file extension

MidnightBlue

keyword: HTML color name



min

keyword: SQL reserved word (z/OS)

MINI-UNIX

operating system

Proprietary OS by Bell Labs.

A version of UNIX Time-Sharing System v6. - Wikipedia: List of operating systems

UNIX Time-Sharing System v6

MintCream

keyword: HTML color name



minus

keyword: SQL reserved word (Oracle)

minute

keyword: SQL reserved word (z/OS)

mlslabel

keyword: SQL reserved word (Oracle)

MistyRose

keyword: HTML color name



MkLinux

operating system

Proprietary OS by Apple for the Apple Macintosh.

Apple Macintosh

See: .mkv 🛍 — file extension

MLYACC

abbreviation

programming language

6

MLYACC - Implementation and output in SML/NJ ftp://research.att.com/dist/ml/75.tools.tar.Z

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

yacc

Moccasin

keyword: HTML color name

Moccasin #FFE4B5

mod

keyword: Ada reserved key word

keyword: VHDL reserved key word

mode

keyword: COBOL reserved word (z/OS)

keyword: SQL reserved word (Oracle)

modify

304

keyword: SQL reserved word (Oracle)

module

keyword: Ruby reserved word

modules

keyword: COBOL reserved word (z/OS)

month

keyword: SQL reserved word (z/OS)

more-labels

keyword: COBOL reserved word (z/OS)

MOS

abbreviation

operating system

Proprietary OS by Acorn Computers

See: .mov 🥝 — file extension

move

keyword: COBOL reserved word (z/OS)

See: .mp3

Image: The extension

See: .mp4 📿 — file extension

See: .mpa — file extension

MP/AOS

abbreviation

operating system

Proprietary OS by Data General.

For Data General micorNOVA-based computers. - Wikipedia: List of operating systems

See: .mpeg 🗟 — file extension

See: .mpg 🗟 — file extension

MP/M

abbreviation

operating system

Proprietary OS by Digital Research, Inc.

Multi-user version of CP/M-80 - Wikipedia: List of operating systems

CP/M

MP/M II

abbreviation

operating system

Proprietary OS by Digital Research, Inc.

Multi-user version of CP/M-80 - Wikipedia: List of operating systems

CP/M

MP/M 8-16

abbreviation

operating system

Proprietary OS by Digital Research, Inc.

a dual-processor variant of MP/M for 8086 and 8080 CPUs. ---Wikipedia: List of operating systems

CP/M

MP/M-86

abbreviation

operating system

Proprietary OS by Digital Research, Inc.

Multi-user version of CP/M-86 - Wikipedia: List of operating systems

CP/M

See: .mrw 🛅 — file extension

See: .msg — file extension

MSP

abbreviation

operating system

Proprietary OS by Fujitsu

MSP-EX

abbreviation

operating system

Proprietary OS by Fujitsu

See: .mts 🏼 — file extension

multiple

keyword: COBOL reserved word (z/OS)

multiply

keyword: COBOL reserved word (z/OS)

Multiuser DOS

operating system

Proprietary OS by Digital Research, Inc.

the successor of Concurrent DOS 386 - Wikipedia: List of operating systems

309

Milo's Dictionary (with Programming Lexicons)

Nn

N

ASCII: 4E hex

UNICODE: LATIN CAPITAL LETTER N

n

ASCII: 6E hex

UNICODE: LATIN SMALL LETTER N

name

keyword: JavaScript object property

namespace

keyword: PHP reserved word

See: _NAMESPACE_ — PHP

NaN

abbreviation

keyword: JavaScript property

nand

keyword: VHDL reserved key word

native

keyword: COBOL reserved word (z/OS) keyword: Java reserved key word

keyword: JavaScript reserved word

native-binary

keyword: COBOL reserved word (z/OS)

natural

keyword: SQL reserved word (z/OS)

NavajoWhite

keyword: HTML color name



navigate

keyword: JavaScript windows method

Navigator

keyword: Navigator (initial cap) JavaScript windows object

Navy

keyword: HTML color name



nchar

keyword: SQL reserved word (z/OS)

See: .nef 🛅 — file extension

negative

keyword: COBOL reserved word (z/OS)

nested_loop

keyword: SQL reserved word (z/OS) 312

new

keyword: Ada reserved key word

keyword: Java reserved key word

keyword: JavaScript reserved word

keyword: PHP reserved word

keyword: VHDL reserved key word

Newton OS

operating system

Proprietary OS by Apple for Apple MessagePad.

next

keyword: COBOL reserved word (z/OS)

keyword: Ruby reserved word

keyword: VHDL reserved key word

nil

keyword: Objective-C word

keyword: Ruby reserved word

no

313

keyword: COBOL reserved word (z/OS)

keyword: NO (all caps) Objective-C word

noaudit

keyword: SQL reserved word (Oracle)

no_cache

keyword: SQL reserved word (z/OS)

nocompress

keyword: SQL reserved word (Oracle)

nonatomic

keyword: Objective-C word

nor

keyword: VHDL reserved key word

NOS (Network Operating System)

abbreviation

operating system

Proprietary OS by Control Data Corporation.

NOS/BE NOS Batch Environemnt

abbreviation

operating system

Proprietary OS by Control Data Corporation.

Ran on the 6x00 and related Cyber. —<u>Wikipedia: List of operating systems</u>

SCOPE

NOS/VE NOS Virtual Environment

abbreviation

operating system

Proprietary OS by Control Data Corporation.

not

keyword: Ada reserved key word

keyword: COBOL reserved word (z/OS)

keyword: Ruby reserved word

keyword: SQL reserved word (ANSI, Oracle, z/OS)

keyword: VHDL reserved key word

noteq

keyword: SQL reserved word (z/OS)

note-off

keyword: MIDI comamnd

8n hex (where n is the channel number)

2 paraaemters

paramater 1: key

paramater 2: velocity

note-on

keyword: MIDI comamnd 9n hex (where n is the channel number) 2 paraaemters paramater 1: key paramater 2: velocity

Novell DOS 7

operating system

Proprietary OS by Digital Research, Inc.

Derived from DR-DOS 3.31-6.0

a single-user, multi-tasking system derived from DR DOS - Wikipedia: List of operating systems

DOS

DR-DOS

Novell PalmDOS 1.0

operating system

Proprietary OS by Digital Research, Inc.

Derived from DR-DOS 3.31-6.0 - Wikipedia: List of operating systems

DOS

DR-DOS

Novell "Star Trek"

operating system

Proprietary OS by Digital Research, Inc.

Derived from DR-DOS 3.31-6.0 - Wikipedia: List of operating systems

DOS

DR-DOS

nowait

keyword: SQL reserved word (Oracle)

null

keyword: Ada reserved key word

keyword: COBOL reserved word (z/OS)

keyword: Java reserved literal value word

keyword: JavaScript reserved word

keyword: NULL (all caps) Objective-C word

keyword: SQL reserved word (ANSI, Oracle, z/OS)

keyword: VHDL reserved key word

nulls

keyword: COBOL reserved word (z/OS)

number

keyword: COBOL reserved word (z/OS) keyword: Number (initial cap) JavaScript object keyword: SQL reserved word (Oracle)

numeric

keyword: COBOL reserved word (z/OS)

keyword: SQL reserved word (z/OS)

numeric-edited

keyword: COBOL reserved word (z/OS)

Oo

0

ASCII: 4F hex

UNICODE: LATIN CAPITAL LETTER O

0

ASCII: 6F hex

UNICODE: LATIN SMALL LETTER O

See: .obj — file extension

object

keyword: COBOL reserved word (z/OS)

object-computer

keyword: COBOL reserved word (z/OS)

Object

keyword: JavaScript object

occurs

keyword: COBOL reserved word (z/OS)

octet_length

keyword: SQL reserved word (z/OS)

See: .odg 🚺 — file extension

See: .ods — file extension

See: .odt — file extension

of

keyword: Ada reserved key word

keyword: COBOL reserved word (z/OS)

keyword: SQL reserved word (ANSI, Oracle, z/OS)

keyword: VHDL reserved key word

off

keyword: COBOL reserved word (z/OS)

Edition 0.35

offline

keyword: SQL reserved word (Oracle)

offscreenBuffering

keyword: JavaScript windows property



OldLace

keyword: HTML color name



Olive

keyword: HTML color name



OliveDrab



omitted

keyword: COBOL reserved word (z/OS)

on

keyword: COBOL reserved word (z/OS)

keyword: SQL reserved word (z/OS)

keyword: VHDL reserved key word

onblur

keyword: JavaScript event handler

onclick

keyword: JavaScript event handler

onerror

keyword: JavaScript event handler

322

oneway

keyword: Objective-C former reserved word

onfocus

keyword: JavaScript event handler

onkeydown

keyword: JavaScript event handler

onkeypress

keyword: JavaScript event handler

onkeyup

keyword: JavaScript event handler

onload

keyword: JavaScript event handler

onmousedown

keyword: JavaScript event handler

onmouseover

keyword: JavaScript event handler

onmouseup

keyword: JavaScript event handler

onsubmit

keyword: JavaScript event handler

open

keyword: COBOL reserved word (z/OS)

keyword: JavaScript windows method

keyword: VHDL reserved key word

opener

keyword: JavaScript windows property

OpenVMS

operating system

Proprietary OS by Digital Equipment Corporation (DEC). Later owned by Compaq, Hewlett-Packard, and Hewlett Packard Enterprise.

alternate name for VMS
VMS

option

keyword: Option (initial cap) JavaScript windows object

keyword: SQL reserved word (ANSI, Oracle)

optional

keyword: COBOL reserved word (z/OS)

See: .opus — file extension

or

keyword: Ada reserved key word

keyword: COBOL reserved word (z/OS)

keyword: PHP reserved word

keyword: Ruby reserved word

keyword: SQL reserved word (ANSI, Oracle, z/OS)

keyword: VHDL reserved key word

Orange

keyword: HTML color name



OrangeRed

keyword: HTML color name



or_bits

keyword: SQL reserved word (z/OS)

Orchid

keyword: HTML color name



order

keyword: COBOL reserved word (z/OS)

keyword: SQL reserved word (ANSI, Oracle, z/OS)

ordered

See: .dotorf 🔨 — file extension

organization

keyword: COBOL reserved word (z/OS)

See: .ost — file extension

OS/8

abbreviation

operating system

Proprietary OS by Digital Equipment Corporation (DEC). Later owned by Compaq, Hewlett-Packard, and Hewlett Packard Enterprise

OSE

abbreviation

operating system

Proprietary OS by ENEA AB

Flexible, small footprint, high-performance RTOS for control processors —<u>Wikipedia: List of</u> operating systems

OS/IV

abbreviation

operating system

Proprietary OS by Fujitsu

OS X

operating system

Proprietary OS by Apple for the Apple Macintosh.

Apple Macintosh

other

keyword: COBOL reserved word (z/OS)

others

keyword: Ada reserved key word

keyword: VHDL reserved key word

out

keyword: Ada reserved key word

keyword: Objective-C former reserved word

keyword: VHDL reserved key word

outer

keyword: SQL reserved word (z/OS)

outerHeight

keyword: JavaScript windows method

outerWidth

keyword: JavaScript windows method

output

keyword: COBOL reserved word (z/OS)

overflow

keyword: COBOL reserved word (z/OS)

overlaps

keyword: SQL reserved word (z/OS)

override

keyword: COBOL reserved word (z/OS)

overriding

keyword: Ada reserved key word

Pp

P

ASCII: 50 hex

UNICODE: LATIN CAPITAL LETTER P

р

ASCII: 70 hex UNICODE: LATIN SMALL LETTER P

package

keyword: Ada reserved key word

keyword: Java reserved key word

keyword: JavaScript reserved word

keyword: VHDL reserved key word

packed-decimal

padding

keyword: COBOL reserved word (z/OS)

page

keyword: COBOL reserved word (z/OS)

page-counter

keyword: COBOL reserved word (z/OS)

pageXOffset

keyword: JavaScript windows method

pageYOffset

keyword: JavaScript windows method

PaleGoldenRod

keyword: HTML color name



PaleGreen

keyword: HTML color name



PaleTurquoise

keyword: HTML color name



PaleVioletRed

keyword: HTML color name

PaleVioletRed #DB7093

PapayaWhip

keyword: HTML color name

PapayaWhip #FFEFD5

parent

keyword: JavaScript windows method

parseFloat

keyword: JavaScript windows method

ParseInt

keyword: JavaScript windows method

password

keyword: COBOL reserved word (z/OS)

See: .pcd 🤯 — file extension

See: .pcm — file extension

pctfree

keyword: SQL reserved word (Oracle)

See: .pcx 🚏 — file extension

See: .pdf — file extension

PeachPuff

keyword: HTML color name



perform

keyword: COBOL reserved word (z/OS)

perl-byacc

programming language

perl-byacc - produces perl output ftp://ftp.sterling.com/local/perl-byacc1.8.2.tar.Z

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

yacc

Personal CP/M

abbreviation

operating system

Proprietary OS by Digital Research, Inc.

A refinement of CP/M for Intel 8080/8085 and Zilog Z80. - Wikipedia: List of operating systems

CP/M

Personal CP/M-86

abbreviation

operating system

Proprietary OS by Digital Research, Inc.

A refinement of CP/M for the Intel 8088/8086 — Wikipedia: List of operating systems

CP/M

Peru

keyword: HTML color name



pf

keyword: COBOL reserved word (z/OS)

ph

keyword: COBOL reserved word (z/OS)

See: .pic — file extension

pic

keyword: COBOL reserved word (z/OS)

335

picture

keyword: COBOL reserved word (z/OS)

Pink

operating system

Proprietary OS by Apple for the Apple Macintosh.

Apple Macintosh

Pink

keyword: HTML color name



pitch bend

keyword: MIDI comamnd

En hex (where n is the channel number)

2 paraaemters

paramater 1: lsb

paramater 2: msb

See: .pkg — file extension

Plan 9 from Bell Labs

Milo's Dictionary (with Programming Lexicons)

operating system

Proprietary OS by Bell Labs.

Plankalküll

programming language

year created: 1943-1945 (published 1948)

created by: Konrad Zuse

predecessors: none (unique language)

Plum

keyword: HTML color name



plus

keyword: COBOL reserved word (z/OS)



pointer

keyword: COBOL reserved word (z/OS)

port

337

keyword: VHDL reserved key word

position

keyword: COBOL reserved word (z/OS)

keyword: SQL reserved word (z/OS)

position_b

keyword: SQL reserved word (z/OS)

position_mb

keyword: SQL reserved word (z/OS)

positive

keyword: COBOL reserved word (z/OS)

postponed

keyword: VHDL reserved key word

PowderBlue

keyword: HTML color name



#B0E0E6

See: .ppt — file extension

See: .pptx — file extension

pragma

keyword: Ada reserved key word

precision

keyword: SQL reserved word (z/OS)

print

keyword: PHP reserved word

printing

keyword: COBOL reserved word (z/OS)

prior

keyword: SQL reserved word (ANSI, Oracle)

private

keyword: Ada reserved key word

keyword: Java reserved key word

keyword: JavaScript reserved word

keyword: PHP reserved word

See: @private — Objective C

privileges

keyword: SQL reserved word (ANSI, Oracle)

procedure

keyword: Ada reserved key word

keyword: COBOL reserved word (z/OS)

keyword: VHDL reserved key word

procedure-pointer

keyword: COBOL reserved word (z/OS)

procedures

keyword: COBOL reserved word (z/OS)

proceed

processing

keyword: COBOL reserved word (z/OS)

Apple ProDOS

operating system

Proprietary OS by Apple for the Apple II family

program

keyword: COBOL reserved word (z/OS)

program change

keyword: MIDI comamnd

Cn hex (where n is the channel number)

1 paraaemter

paramater: instrument number

program-id

keyword: COBOL reserved word (z/OS)

prompt

keyword: JavaScript windows method

See: @property — Objective C

propertyIsEnum

keyword: JavaScript windows method

protected

keyword: Ada reserved key word

keyword: Java reserved key word

keyword: JavaScript reserved word

keyword: PHP reserved word

See: @protected — Objective C

Protocol

keyword: Objective-C word

See: @protocol — OBJECTIVE C

prototype

keyword: JavaScript object



See: .psdx — file extension

See: .pst — file extension

See: .ptx — file extension

public

keyword: Java reserved key word

keyword: JavaScript reserved word

keyword: PHP reserved word

keyword: SQL reserved word (ANSI, Oracle)

See: @public — OBJECTIVE C

Puffin OS

operating system

Proprietary OS by CloudMosa.

pure

keyword: VHDL reserved key word

purge

keyword: COBOL reserved word (z/OS)

Purple

343



PWB/UNIX

abbreviation

operating system

Proprietary OS by Bell Labs.

A version of UNIX Time-Sharing System v6. - Wikipedia: List of operating systems

UNIX Time-Sharing System v6

See: .pzl 🤯 — file extension

Qq

Q

ASCII: 51 hex

UNICODE: LATIN CAPITAL LETTER Q

q

ASCII: 71 hex

UNICODE: LATIN SMALL LETTER Q

queue

keyword: COBOL reserved word (z/OS)

quote

keyword: COBOL reserved word (z/OS)

quotes

Rr

R

ASCII: 52 hex

UNICODE: LATIN CAPITAL LETTER R

r

ASCII: 72 hex

UNICODE: LATIN SMALL LETTER R

See: .raf ī — file extension

raise

keyword: Ada reserved key word

random

range

keyword: Ada reserved key word

keyword: VHDL reserved key word

See: .rar — file extension

See: .raw — file extension

raw

keyword: SQL reserved word (Oracle)

rd

keyword: COBOL reserved word (z/OS)

RDOS

abbreviation

operating system

Proprietary OS by Data General.

real-time Disk Operating System, with variants: RTOS and DOS. —<u>Wikipedia: List of operating</u> systems

read

ready

keyword: COBOL reserved word (z/OS)

real

keyword: SQL reserved word (z/OS)

RebeccaPurple

keyword: HTML color name

See: .rec — file extension

receive

keyword: COBOL reserved word (z/OS)

record

keyword: Ada reserved key word

keyword: COBOL reserved word (z/OS)

keyword: VHDL reserved key word

recording

records

keyword: COBOL reserved word (z/OS)

recursive

keyword: COBOL reserved word (z/OS)

Red

keyword: HTML color name



redefines

keyword: COBOL reserved word (z/OS)

redo

keyword: Ruby reserved word

reel

See: .ref — file extension

reference

keyword: COBOL reserved word (z/OS)

references

keyword: COBOL reserved word (z/OS)

register

keyword: Objective-C reserved word

keyword: VHDL reserved key word

reject

keyword: VHDL reserved key word

See: .rel — file extension

relative

keyword: COBOL reserved word (z/OS)

release

reload

keyword: COBOL reserved word (z/OS)

rem

keyword: Ada reserved key word

keyword: VHDL reserved key word

remainder

keyword: COBOL reserved word (z/OS)

removal

keyword: COBOL reserved word (z/OS)

rename

keyword: SQL reserved word (Oracle)

renames

keyword: Ada reserved key word

keyword: COBOL reserved word (z/OS)

replace

352

keyword: COBOL reserved word (z/OS)

replacing

keyword: COBOL reserved word (z/OS)

report

keyword: COBOL reserved word (z/OS)

keyword: VHDL reserved key word

reporting

keyword: COBOL reserved word (z/OS)

reports

keyword: COBOL reserved word (z/OS)

repository

keyword: COBOL reserved word (z/OS)

requeue

keyword: Ada reserved key word

353

required

keyword: PHP reserved word

required_once

keyword: PHP reserved word

rerun

keyword: COBOL reserved word (z/OS)

rescue

keyword: Ruby reserved word

reserve

keyword: COBOL reserved word (z/OS)

reset

keyword: COBOL reserved word (z/OS)

keyword: JavaScript windows method

resource

keyword: SQL reserved word (Oracle)

restrict

keyword: Objective-C reserved word

retain

keyword: Objective-C word

retry

keyword: Ruby reserved word

return

keyword: Ada reserved key word keyword: COBOL reserved word (z/OS) keyword: Java reserved key word keyword: JavaScript reserved word keyword: Objective-C reserved word keyword: PHP reserved word keyword: Ruby reserved word keyword: VHDL reserved key word

return-code

returning

keyword: COBOL reserved word (z/OS)

reverse

keyword: Ada reserved key word

reversed

keyword: COBOL reserved word (z/OS)

revoke

keyword: SQL reserved word (ANSI, Oracle)

rewind

keyword: COBOL reserved word (z/OS)

rewrite

keyword: COBOL reserved word (z/OS)

rf

rh

keyword: COBOL reserved word (z/OS)

Rhapsody

operating system

Proprietary OS by Apple for the Apple Macintosh.

Apple Macintosh

right

keyword: COBOL reserved word (z/OS)

keyword: SQL reserved word (z/OS)

RISC iX

abbreviation

operating system

Proprietary OS by Acorn Computers

RISC OS

abbreviation

operating system

Proprietary OS by Acorn Computers

See: .rmvb 쭏 — file extension

rol

keyword: VHDL reserved key word

ror

keyword: VHDL reserved key word

RosyBrown

keyword: HTML color name



round

keyword: SQL reserved word (z/OS)

rounded

keyword: COBOL reserved word (z/OS)

row

keyword: SQL reserved word (Oracle, z/OS)

rowid

keyword: SQL reserved word (Oracle)

rownum

keyword: SQL reserved word (Oracle)

rows

keyword: SQL reserved word (ANSI, Oracle)

RoyalBlue

keyword: HTML color name



rpad

keyword: SQL reserved word (z/OS)

See: .rpm — file extension

RSTS/E

abbreviation

operating system

Proprietary OS by Digital Equipment Corporation (DEC). Later owned by Compaq, Hewlett-Packard, and Hewlett Packard Enterprise

multi-user time-sharing OS for PDP-11s - Wikipedia: List of operating systems

RSX-11

abbreviation

operating system

Proprietary OS by Digital Equipment Corporation (DEC). Later owned by Compaq, Hewlett-Packard, and Hewlett Packard Enterprise

multiuser, multitasking OS for PDP-11s - Wikipedia: List of operating systems

RT-11

abbreviation

operating system

Proprietary OS by Digital Equipment Corporation (DEC). Later owned by Compaq, Hewlett-Packard, and Hewlett Packard Enterprise

single user OS for PDP-11 - Wikipedia: List of operating systems

run

keyword: COBOL reserved word (z/OS)

See: .rw2 — file extension
55

S

ASCII: 53 hex

UNICODE: LATIN CAPITAL LETTER S

S

ASCII: 73 hex

UNICODE: LATIN SMALL LETTER S

SaddleBrown

keyword: HTML color name

SaddleBrown #8B4513

Salmon

keyword: HTML color name



same

keyword: COBOL reserved word (z/OS)

same_source

keyword: SQL reserved word (z/OS)

SandyBrown

keyword: HTML color name



SASL-yacc

abbreviation

programming language

SASL-yacc - "Yacc in SASL - An Exercise in Functional Programming", Simon Peyton-Jones, Software Prac & Exp 15:807-820 (1985). Mentions also a BCPL implementation.

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas yacc

See: .sav — file extension

SCOPE

operating system

Proprietary OS by Control Data Corporation.

SCOPE (Supervisory Control Of Program Execution ran on the Lower 3000 series, Upper 3000 series, and 6x00 and rleated Cyber. —<u>Wikipedia: List of operating systems</u>

screenX

keyword: JavaScript windows method

screenY

keyword: JavaScript windows method

scroll

keyword: JavaScript windows method

sd

keyword: COBOL reserved word (z/OS)

See: .sda 🜌 — file extension

SeaGreen

keyword: HTML color name



search

keyword: COBOL reserved word (z/OS)

SeaShell

keyword: HTML color name



second

keyword: SQL reserved word (z/OS)

section

keyword: COBOL reserved word (z/OS)

security

segment

keyword: COBOL reserved word (z/OS)

segment-limit

keyword: COBOL reserved word (z/OS)

SEL

keyword: Objective-C word

select

keyword: Ada reserved key word

keyword: COBOL reserved word (z/OS)

keyword: Select (initial cap) JavaScript windows object

keyword: SQL reserved word (ANSI, Oracle, z/OS)

keyword: VHDL reserved key word

See: @select — OBJECTIVE C

self

keyword: COBOL reserved word (z/OS)

keyword: JavaScript windows property

keyword: Objective-C reserved word

keyword: Ruby reserved word

send

keyword: COBOL reserved word (z/OS)

sentence

keyword: COBOL reserved word (z/OS)

seperate

keyword: Ada reserved key word

keyword: COBOL reserved word (z/OS)

sequence

keyword: COBOL reserved word (z/OS)

sequential

keyword: COBOL reserved word (z/OS)

sequentially

keyword: SQL reserved word (z/OS)

366

service

keyword: COBOL reserved word (z/OS)

session

keyword: SQL reserved word (ANSI, Oracle)

set

keyword: COBOL reserved word (z/OS)

keyword: SQL reserved word (ANSI, Oracle, z/OS)

setInterval

keyword: JavaScript windows method

setTimeout

keyword: JavaScript windows method

severity

keyword: VHDL reserved key word

share

keyword: SQL reserved word (Oracle)

operating system

Proprietary OS by IBM

for the IBM 704 and 709 - Wikipedia: List of operating systems

shift-in

keyword: COBOL reserved word (z/OS)

shift-out

keyword: COBOL reserved word (z/OS)

short

keyword: Java reserved key word

keyword: JavaScript reserved word

keyword: Objective-C reserved word

Siemens S5-DOS/MT

abbreviation

operating system

Proprietary OS by Digital Research, Inc.

an industrial control system based on FlexOS - Wikipedia: List of operating systems

Sienna

keyword: HTML color name



sign

keyword: COBOL reserved word (z/OS)

signal

keyword: VHDL reserved key word

signed

keyword: Objective-C reserved word

Silver

keyword: HTML color name

Silver #C0C0C0

simple_nested_loop

369

keyword: SQL reserved word (z/OS)

SIPROS (Simultaneous Processing Operating System)

operating system

Proprietary OS by Control Data Corporation.

shared

keyword: VHDL reserved key word

SkyBlue

keyword: HTML color name



size

keyword: COBOL reserved word (z/OS)

keyword: SQL reserved word (ANSI, Oracle)

skip1

skip2

keyword: COBOL reserved word (z/OS)

skip3

keyword: COBOL reserved word (z/OS)

sla

keyword: VHDL reserved key word

SlateBlue

keyword: HTML color name



SlateGray

keyword: HTML color name



SlateGrey

keyword: HTML color name



sll

keyword: VHDL reserved key word

sizeof

keyword: Objective-C reserved word

smallint

keyword: SQL reserved word (ANSI, Oracle, z/OS)

See: .snd — file extension

See: .sng — file extension

Snow

keyword: HTML color name



sort

sort-control

keyword: COBOL reserved word (z/OS)

sort-core-size

keyword: COBOL reserved word (z/OS)

sort-file-size

keyword: COBOL reserved word (z/OS)

sort-merge

keyword: COBOL reserved word (z/OS)

sort-message

keyword: COBOL reserved word (z/OS)

sort-mode-size

keyword: COBOL reserved word (z/OS)

sort-return

See: SOS — operating system

source

keyword: COBOL reserved word (z/OS)

source-computer

keyword: COBOL reserved word (z/OS)

space

keyword: COBOL reserved word (z/OS)

spaces

keyword: COBOL reserved word (z/OS)

special-names

keyword: COBOL reserved word (z/OS)

SpringGreen

keyword: HTML color name



See: .sql — file extension

sra

keyword: VHDL reserved key word

srl

keyword: VHDL reserved key word

standard

keyword: COBOL reserved word (z/OS)

standard-1

keyword: COBOL reserved word (z/OS)

standard-2

keyword: COBOL reserved word (z/OS)

start

keyword: COBOL reserved word (z/OS)

keyword: SQL reserved word (Oracle)

static

375

keyword: Java reserved key word

keyword: JavaScript reserved word

keyword: Objective-C reserved word

keyword: PHP reserved word

status

keyword: COBOL reserved word (z/OS)

keyword: JavaScript windows property

stdout

keyword: SQL reserved word (z/OS)

SteelBlue

keyword: HTML color name



stop

keyword: COBOL reserved word (z/OS)

stream ASCII file format

PDP-11 file format for plain text ASCII files

Source files, such as MACROĐ11 and FORTRAN IV programs, and text files that you create with an editor are in stream ASCII format. These files consist of a series of bytes, each byte representing an ASCII character. Stream ASCII files have no special headers or end blocks, nor do they include any formatted binary blocks.

—RT-11 Volume and File Formats Manual:

strictfp

keyword: Java reserved key word

string

keyword: COBOL reserved word (z/OS)

keyword: String (initial cap) JavaScript object

struct

keyword: Objective-C reserved word

subarray

keyword: SQL reserved word (z/OS)

submit

keyword: JavaScript windows method

sub-queue-1

keyword: COBOL reserved word (z/OS)

sub-queue-2

keyword: COBOL reserved word (z/OS)

sub-queue-3

keyword: COBOL reserved word (z/OS)

substring_mb

keyword: SQL reserved word (z/OS)

substring

keyword: SQL reserved word (z/OS)

subtract

keyword: COBOL reserved word (z/OS)

subtype

keyword: Ada reserved key word

keyword: VHDL reserved key word

successful

keyword: SQL reserved word (Oracle)

sum

keyword: COBOL reserved word (z/OS)

keyword: SQL reserved word (z/OS)

See: .sup

super

keyword: COBOL reserved word (z/OS)

keyword: Java reserved key word

keyword: JavaScript reserved word

keyword: Super (initial cap) Objective-C reserved word

keyword: Ruby reserved word

suppress

keyword: COBOL reserved word (z/OS)

See: .svg 🛴 — file extension

Milo's Dictionary (with Programming Lexicons)

See: .svgz — file extension

See: .swf 💋 — file extension

switch

keyword: Java reserved key word

keyword: JavaScript reserved word

keyword: Objective-C reserved word

keyword: PHP reserved word

symbolic

keyword: COBOL reserved word (z/OS)

SymbOS

operating system

Proprietary OS by Amstrad

sync

keyword: COBOL reserved word (z/OS)

synchronized

keyword: Ada reserved key word

keyword: COBOL reserved word (z/OS)

keyword: Java reserved key word ³⁸⁰ keyword: JavaScript reserved word

synonym

keyword: SQL reserved word (Oracle)

See: @synthesize — OBJECTIVE C

sysdate

keyword: SQL reserved word (Oracle)

Tt

T

ASCII: 54 hex

UNICODE: LATIN CAPITAL LETTER T

t

ASCII: 74 hex

UNICODE: LATIN SMALL LETTER T

table

keyword: COBOL reserved word (z/OS)

keyword: SQL reserved word (ANSI, Oracle, z/OS)

tagged

keyword: Ada reserved key word

taint

keyword: JavaScript windows depreciated function

tally

keyword: COBOL reserved word (z/OS)

tallying

keyword: COBOL reserved word (z/OS)

Tan

keyword: HTML color name



tape

keyword: COBOL reserved word (z/OS)

See: .tar — file extension

See: .tar.gz — file extension

task

keyword: Ada reserved key word

Teal

keyword: HTML color name



TENEX

abbreviation

operating system

Proprietary OS by Digital Equipment Corporation (DEC). Later owned by Compaq, Hewlett-Packard, and Hewlett Packard Enterprise

an ancestor of TOPS-20 from BBN, for the PDP-10 - Wikipedia: List of operating systems

terminal

keyword: COBOL reserved word (z/OS)

terminate

keyword: Ada reserved key word

keyword: COBOL reserved word (z/OS)

test

text

keyword: COBOL reserved word (z/OS)

keyword: JavaScript windows property

See: .tga 🖤 — file extension

than

keyword: COBOL reserved word (z/OS)

then

keyword: Ada reserved key word

keyword: COBOL reserved word (z/OS)

keyword: Ruby reserved word

keyword: SQL reserved word (ANSI, Oracle, z/OS)

keyword: VHDL reserved key word

this

keyword: Java reserved key word

keyword: JavaScript reserved word

Thistle

keyword: HTML color name 385



See: .thm — file extension

through

keyword: COBOL reserved word (z/OS)

throw

keyword: Java reserved key word

keyword: JavaScript reserved word

keyword: PHP reserved word

See: @throw — OBJECTIVE C

throws

keyword: Java reserved key word

keyword: JavaScript reserved word

thru

See: .tiff I extension

time

keyword: COBOL reserved word (z/OS)

keyword: SQL reserved word (z/OS)

times

keyword: COBOL reserved word (z/OS)

timestamp

keyword: SQL reserved word (z/OS)

timesten

keyword: SQL reserved word (z/OS)

title

keyword: COBOL reserved word (z/OS)

See: .tmp — file extension

to

keyword: SQL reserved word (ANSI, Oracle, z/OS)

keyword: VHDL reserved key word



Tomato

keyword: HTML color name



top

keyword: COBOL reserved word (z/OS)

keyword: JavaScript windows property

TOPS-10

abbreviation

operating system

Proprietary OS by Digital Equipment Corporation (DEC). Later owned by Compaq, Hewlett-Packard, and Hewlett Packard Enterprise

for the PDP-10

TOPS-20

abbreviation

operating system

Proprietary OS by Digital Equipment Corporation (DEC). Later owned by Compaq, Hewlett-Packard, and Hewlett Packard Enterprise

for the PDP-10 — Wikipedia: List of operating systems

Toshiba 4690 OS

operating system

Proprietary OS by Digital Research, Inc.

a Point of Sale (POS) operating system based on IBM 4690 OS and FlexOS

DOS

toString

keyword: JavaScript method

Towns OS

operating system

Proprietary OS by Fujitsu

See: .tp 📟 — file extension

trace

trailing

keyword: COBOL reserved word (z/OS)

keyword: SQL reserved word (z/OS)

trait

keyword: PHP reserved word

See: _TRAIT_ — PHP

transient

keyword: Java reserved key word

keyword: JavaScript reserved word

transport

keyword: VHDL reserved key word

trigger

keyword: SQL reserved word (Oracle)

trim

trim_b

keyword: SQL reserved word (z/OS)

trim_mb

keyword: SQL reserved word (z/OS)

Tru64 UNIX

operating system

Proprietary OS by Digital Equipment Corporation (DEC). Later owned by Compaq, Hewlett-Packard, and Hewlett Packard Enterprise.

alternate name for Digital Unix

Digital UNIX

true

keyword: COBOL reserved word (z/OS)

keyword: Java reserved literal value word

keyword: JavaScript reserved word

keyword: Ruby reserved word

keyword: SQL reserved word (z/OS)

try

keyword: Java reserved key word

keyword: JavaScript reserved word

keyword: PHP reserved word

See: @try — OBJECTIVE C

See: .ts - file extension

Turquoise

keyword: HTML color name



tvOS

abbreviation

operating system

Proprietary OS by Apple for Apple TV.

See: .txt — file extension

type

keyword: Ada reserved key word

keyword: COBOL reserved word (z/OS)

keyword: VHDL reserved key word

typedef

keyword: Objective-C reserved word

typeof

keyword: JavaScript reserved word

Uu

U

ASCII: 55 hex

UNICODE: LATIN CAPITAL LETTER U

u

ASCII: 75 hex

UNICODE: LATIN SMALL LETTER U

See: .uax 🥨 — file extension

uid

keyword: SQL reserved word (Oracle)

Ultrix

operating system

Proprietary OS by Digital Equipment Corporation (DEC). Later owned by Compaq, Hewlett-Packard, and Hewlett Packard Enterprise

394

See: UMES — operating system

unaffected

keyword: VHDL reserved key word

undef

keyword: Ruby reserved word

undefined

keyword: JavaScript property

unescape

keyword: JavaScript windows depreciated function

union

keyword: Objective-C reserved word

keyword: SQL reserved word (ANSI, Oracle, z/OS)

unique

keyword: SQL reserved word (ANSI, Oracle)

unit

keyword: COBOL reserved word (z/OS)

units

keyword: VHDL reserved key word

University of Michigan Executive System (UMES)

operating system

Proprietary OS by IBM

for the IBM 704, 709, and 7090 - Wikipedia: List of operating systems

Unix System III

operating system

Proprietary OS by Bell Labs.

A version of UNIX Time-Sharing System v7. - Wikipedia: List of operating systems

UNIX Time-Sharing System v7

Unix System IV

operating system

Proprietary OS by Bell Labs.

A version of UNIX Time-Sharing System v7. - Wikipedia: List of operating systems

UNIX Time-Sharing System v7
Unix System V

operating system

Proprietary OS by Bell Labs.

A version of UNIX Time-Sharing System v7. Releases include 2.0, 3.0, 3.2, 4.0, and 4.2. —<u>Wikipedia: List of operating systems</u>

UNIX Time-Sharing System v7

UNIX Time-Sharing System v1

operating system

Proprietary OS by Bell Labs.

UNIX Time-Sharing System v2

operating system

Proprietary OS by Bell Labs.

UNIX Time-Sharing System v3

operating system

Proprietary OS by Bell Labs.

UNIX Time-Sharing System v4

operating system

Proprietary OS by Bell Labs.

UNIX Time-Sharing System v5

operating system

Proprietary OS by Bell Labs.

UNIX Time-Sharing System v6

operating system

Proprietary OS by Bell Labs.

Versions include MINI-UNIX, PWB/UNIX, USG, and CB Unix. —<u>Wikipedia: List of operating</u> systems

UNIX Time-Sharing System v7

operating system

Proprietary OS by Bell Labs.

Includes Unix System III, Unix System IV, and Unix System V. Almost all modern versions of UNIX are derived from this version. —<u>Wikipedia: List of operating systems</u>

UNIX Time-Sharing System v8

operating system

Proprietary OS by Bell Labs.

UNIX Time-Sharing System v9

operating system

Proprietary OS by Bell Labs.

UNIX Time-Sharing System v10

operating system

Proprietary OS by Bell Labs.

unknown

keyword: SQL reserved word (z/OS)

unless

keyword: Ruby reserved word

[unnamed]

operating system

Unnamed proprietary embedded operating system (NetBSD variant) by Apple for the Airport Extreme and Time Capsule. —<u>Wikipedia: List of operating systems</u>

unset()

keyword: PHP reserved word (this is a reserved word and not a function)

unsigned

keyword: Objective-C reserved word

unstring

keyword: COBOL reserved word (z/OS)

untaint

keyword: JavaScript windows depreciated function

until

keyword: Ada reserved key word keyword: COBOL reserved word (z/OS) keyword: Ruby reserved word keyword: VHDL reserved key word

up

keyword: COBOL reserved word (ANSI, Oracle, z/OS)

update

keyword: SQL reserved word (z/OS)

upon

keyword: COBOL reserved word (z/OS)

upper

keyword: SQL reserved word (z/OS)

upper_mb

keyword: SQL reserved word (z/OS)

usage

keyword: COBOL reserved word (z/OS)

use

keyword: Ada reserved key word keyword: COBOL reserved word (z/OS) keyword: PHP reserved word keyword: VHDL reserved key word

user

keyword: SQL reserved word (ANSI, Oracle)

USG

abbreviation

operating system

Proprietary OS by Bell Labs.

A version of UNIX Time-Sharing System v6. - Wikipedia: List of operating systems

UNIX Time-Sharing System v6

using

keyword: COBOL reserved word (z/OS)

keyword: SQL reserved word (z/OS)

V

ASCII: 56 hex

UNICODE: LATIN CAPITAL LETTER V

V

ASCII: 76 hex

UNICODE: LATIN SMALL LETTER V

validate

keyword: SQL reserved word (Oracle)

value

keyword: COBOL reserved word (z/OS)

valueOf

keyword: JavaScript method

values

keyword: COBOL reserved word (z/OS)

keyword: SQL reserved word (ANSI, Oracle, z/OS)

var

keyword: JavaScript reserved word

keyword: PHP reserved word

varchar

keyword: SQL reserved word (ANSI, Oracle, z/OS)

varchar2

keyword: SQL reserved word (ANSI, Oracle)

variable

keyword: VHDL reserved key word

varying

keyword: COBOL reserved word (z/OS)

```
keyword: SQL reserved word (z/OS)
```

See: .vcd — file extension

See: .vcf — file extension

view

keyword: SQL reserved word (ANSI, Oracle)

Violet

keyword: HTML color name



VMS

abbreviation

operating system

Proprietary OS by Digital Equipment Corporation (DEC). Later owned by Compaq, and Hewlett-Packard

Now by VMS Software Inc. for the VAX mini-computer range, Alpha, and Intel Itanium i2 and i4; later renamed OpenVMS —<u>Wikipedia: List of operating systems</u>

See: .vob 📟 — file extension

void

keyword: Java reserved key word

Milo's Dictionary (with Programming Lexicons)

keyword: JavaScript reserved word

keyword: Objective-C reserved word

volatile

keyword: Java reserved key word

keyword: JavaScript reserved word

keyword: Objective-C reserved word

Von Neumann and Goldstine graphing system

programming language

year created: 1946

created by: John von Neumann and Herman Goldstine

predecessors: ENIAC coding system

See: .vsdx

Vulcan O/S

abbreviation

operating system

Proprietary OS by Harris Corporation

for Harris' Computer Systems (HCX) - Wikipedia: List of operating systems

Ww

W

ASCII: 57 hex

UNICODE: LATIN CAPITAL LETTER W

W

ASCII: 77 hex

UNICODE: LATIN SMALL LETTER W

wait

keyword: VHDL reserved key word

WAITS

abbreviation

operating system

Proprietary OS by Digital Equipment Corporation (DEC). Later owned by Compaq, Hewlett-Packard, and Hewlett Packard Enterprise.

for the PDP-6 and PDP-10 - Wikipedia: List of operating systems

watchOS

operating system

Proprietary OS by Apple for Apple Watch.



Wear OS

operating system

Proprietary OS by Google

a version of Google's Android operating system designed for smartwatches and other wearables. -Wikipedia: List of operating systems

WebAssembly

keyword: JavaScript object

web browser

web brau'-zər, noun

Application software for accessing the world wide web.

See: .webm 🖸 — file extension

See: .webp — file extension

Wheat

keyword: HTML color name



when

- keyword: Ada reserved key word
- keyword: COBOL reserved word (z/OS)
- keyword: Ruby reserved word
- keyword: SQL reserved word (z/OS)
- keyword: VHDL reserved key word

when-compiled

keyword: COBOL reserved word (z/OS)

whenever

keyword: SQL reserved word (ANSI, Oracle)

where

keyword: SQL reserved word (ANSI, Oracle, z/OS)

while

- keyword: Ada reserved key word
- keyword: Java reserved key word
- keyword: JavaScript reserved word
- keyword: Objective-C reserved word
- keyword: PHP reserved word
- keyword: Ruby reserved word
- keyword: VHDL reserved key word

White

keyword: HTML color name

White #FFFFFF

WhiteSmoke

keyword: HTML color name



window

keyword: JavaScript windows object

with

keyword: Ada reserved key word

keyword: COBOL reserved word (z/OS)

keyword: JavaScript reserved word

keyword: SQL reserved word (ANSI, Oracle)

keyword: VHDL reserved key word

See: .wma 🗟 — file extension

.wmf

file extension:

Vector graphics: Windows Metafile Format

See: .wmv 🗟 — file extension

words

keyword: COBOL reserved word (z/OS)

working-storage

keyword: COBOL reserved word (z/OS)

world web web

world wide web, noun

An interconnected information system of public webpage accessible through hyperlinks. Invented in 198 by Tim Berners-Lee working at CERN.

See: .wpl — file extension

write

keyword: COBOL reserved word (z/OS)

write-only

keyword: COBOL reserved word (z/OS)

WSFN

programming language

year created: 1983

WSFN - Which Stands For Nothing. Atari 1983. Beginner's language with emphasis on graphics, for Atari home computers. Version: Advanced WSFN.

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

programming language

6

WSL - Waterloo Systems Language. A C-like systems programming language. "Waterloo Systems Language: Tutorial and Language Reference", F.D. Boswell, WATFAC Publications Ltd, Waterloo, Canada. ISBN 0-919884-00-8.

---**The Language List:** information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas



X

ASCII: 58 hex

UNICODE: LATIN CAPITAL LETTER X

X

ASCII: 78 hex

UNICODE: LATIN SMALL LETTER X

X-1

programming language

X-1 - Early system on UNIVAC I or II. Listed in CACM 2(5):16 (May 1959).

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

programming language

Xbase - Generic term for the dBASE family of languages. Coined in response to threatened litigation over use of the trademark "dBASE."

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

XBASIC

programming language

year created: 1972

XBASIC - eXtended BASIC. 1972. An extension of BASIC, including matrix operations and Algol-like procedures. For the Univac 1108.

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

XC

programming language

year created: 1987

XC - Declarative extension of C++. "XC - A Language for Embedded Rule Based Systems", E. Nuutila et al, SIGPLAN Notices 22(9):23-32 (Sep 1987).

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

XDL

programming language

XDL - An object-oriented extension to CCITT's SDL[2]. "XDL: An Object- Oriented Extension to SDL", S.J. Ochuodho et al in [?]

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

Xfun

programming language

year created: 1991

Xfun - S. Dalmas <dalmas@sophia.inria.fr>, INRIA, 1991. A cross between SML and Russell, intended for computer algebra. "A Polymorphic Functional language Applied to Symbolic Computation", S. Dalmas, Proc Intl Symp Symb Alg Comp, Berkeley 1992.

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

Xi

programming language

year created: 1982

Xi - VLSI design language. "The Circuit Design Language Xi", S.I. Feldman, unpublished

memo, Bell Labs, 1982. Mentioned in Computational Aspects of VLSI, J.D. Ullman, CS Press 1984.

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

XICS

programming language

XICS - Xerox. Page description language.

--The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

XL

programming language

year created: 1977

XL - A tuple language used as the intermediate form in the code generator generator XGEN. "A Knowledge Based Code Generator Generator", Proc Symp AI and Prog Lang, Aug 1977, pp.126-129.

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

XLISP

programming language

6

XLISP - eXperimental LISP. David Betz <dbetz@apple.com>. LISP variant with objectoriented extensions, portable source in C. BYTE article. Version 2.0. Version 2.1 by Tom Almy <toma@sail.labs.tek.com> is closer to Common Lisp.

ftp://cs.orst.edu/pub/xlisp/*

//glia.biostr.washington.edu/pub/xlisp/xlisp21e.tar.Z

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

XLISP-PLUS

programming language

XLISP-PLUS - An extension of XLISP used in the WINTERP OSF/Motif Widget interpreter. http://www.eit.com/software/winterp/winterp/html ftp://ftp.x.org/contrib/devel_tools/winterp*

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

See: .xls — file extension

See: .xlsx — file extension

See: .xml — file extension

See: .xmp 🖾 — file extension

XNF

programming language

6

XNF - Hardware description language?

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

xnor

keyword: VHDL reserved key word

xor

keyword: Ada reserved key word

keyword: PHP reserved word

keyword: VHDL reserved key word

xor_bits

keyword: SQL reserved word (z/OS)

XPC

programming language

year created: 1989

XPC - eXplicitly Parallel C. Dialect of Parallel C which is mode independent, i.e. efficiently compilable to both SIMD and MIMD architectures. "Toward Semantic Self-Consistency in Explicitly Parallel Languages," M.J. Phillip & H.G. Dietz, Proc 4th Intl Conf on Supercomputing, Santa Clara, CA, May 1989, v.1, pp.398-407. Research implementations only.

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

XPL

programming language

year created: 1967-1969

XPL - Stanford, 1967-69. Small dialect of PL/I used for compiler writing. One-dimensional arrays. I/O achieved with character pseudo-variable INPUT and OUTPUT, e.g. OUTPUT = 'This is a line'; Inline machine code. "Programmers are given all the rope they ask for. Novices tend to hang themselves fairly frequently." Implemented on IBM 360, Univac 1100, ICL System 4, CDC6000 and Cyber series, XDS Sigma-5 and Sigma-7, and DEC PDP- 10. "A Compiler Generator," W.M. McKeeman et al, P-H 1970. Also JCC, AFIPS 1968.

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

XPOP

programming language

year created: 1964

(

XPOP - Extensible macro assembly language with user-redefinable grammar, for use with FAP. "XPOP: A Meta-language Without Metaphysics", M.I. Halpern, Proc FJCC 25:57-68, AFIPS (Fall 1964).

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

XScheme

programming language

XScheme - David Betz. Scheme with object-oriented extensions. Source in C. Versions for PC, Macintosh, Atari, Amiga. ftp://labrea.stanford.edu/pub/xscheme.tar.gz

comp.sources.amiga/volume90, version 0.28 for Amiga

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

XSP

abbreviation

operating system

Proprietary OS by Fujitsu

XTRAN

programming language

XTRAN - FORTRAN-like, interactive. [?]

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

XTS-400

abbreviation

operating system

Proprietary OS by BAE Systems.



Y

ASCII: 59 hex

UNICODE: LATIN CAPITAL LETTER Y

У

ASCII: 79 hex

UNICODE: LATIN SMALL LETTER Y

Y

programming language

year created: 1981

Y - General purpose language syntactically like RATFOR, semantically like C. Lacks structures and pointers. Used as a source language for the Davidson/Fraser peephole optimizer. "The Y Programming Language", D.R. Hanson, SIGPLAN Notices 16(2):59-68 (Feb 1981). ftp://ftp.cs.princeton.edu/pub/y+po.tar.Z

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

Yaa

abbreviation

programming language

Yaa - Yet Another Assembler - Macro assembler for GCOS 8 and Mark III on Bull DPS-8 machines. Available from Bull as part of U Waterloo Tools package (maintained by <pjf@thinkage.on.ca>).

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

yacc

abbreviation

programming language

year created: 1975

6

yacc - Yet Another Compiler Compiler. Language used by the Yacc LALR parser generator. "YACC - Yet Another Compiler Compiler", S.C. Johnson, CS TR 32, Bell Labs (Jul 1975).

Implementations:

ayacc - UC Irvine. written in Ada, produces Ada output

ftp://liege.ics.uci.edu/pub/irus/aflex-ayacc_1.2a.tar.Z

Bison - from GNU ftp://prep.ai.mit.edu/pub/gnu/bison-1.21.tar.Z

Bison++ - produces C++ output. ftp://psuvax1.cs.psu.edu/pub/src/gnu/bison++-1.04.tar.Z perl-byacc - produces perl output ftp://ftp.sterling.com/local/perl-byacc1.8.2.tar.Z

SASL-yacc - "Yacc in SASL - An Exercise in Functional Programming", Simon Peyton-Jones, Software Prac & Exp 15:807-820 (1985). Mentions also a BCPL implementation.

yacc++ - 1990. An object-oriented rewrite of yacc, supports regular exp- ressions,

produces an LR(1) parser. "YACC Meets C++", S.C. Johnson, USENIX Spring '88 Conf. Chris Clark, Compiler Resources Inc, Barbara Zino <bz%compres.UUCP@primerd.cv.com> (508) 435-5016.

MLYACC - Implementation and output in SML/NJ ftp://research.att.com/dist/ml/75.tools.tar.Z

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

yacc++

abbreviation

programming language

year created: 1990

yacc++ - 1990. An object-oriented rewrite of yacc, supports regular exp- ressions, produces an LR(1) parser. "YACC Meets C++", S.C. Johnson, USENIX Spring '88 Conf. Chris Clark, Compiler Resources Inc, Barbara Zino <bz%compres.UUCP@primerd.cv.com> (508) 435-5016.

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

yacc

YALLL

abbreviation

programming language

year created: 1979

YALLL - Yet Another Low Level Language. Patterson et al, UC Berkeley, 1979. A microprogramming language resembling conventional assembly language. "Towards an Efficient Machine-Independent Language or Microprogramming", D.A. Patterson et al, Proc 12th Ann Workshop Microprogramming (MICRO-12), 1979, pp.22-35

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

YAPS

abbreviation

programming language

YAPS - Yet Another Production System? College Park Software. Commercial production rule language, simpler than OPS5. Allows knowledge bases to be attached to instances of CLOS objects.

info: Liz Allen liz@grian.cps.altadena.cs.us>

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

YASOS

abbreviation

programming language

YASOS - Yet Another Scheme Object System. info: Ken Dickey <kend@newton.apple.com>

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

Yay

abbreviation

programming language

Yay - Yet Another Yacc - An extension of Yacc with LALR(2) parsing. Available from Bull as part of U Waterloo Tools package (maintained by <pjf@thinkage.on.ca>).

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

year

keyword: SQL reserved word (z/OS)

Yellow

programming language

year created: 1978

Yellow - SRI. A language proposed to meet the Ironman requirements which led to Ada. "On the YELLOW Language Submitted to the DoD", E.W. Dijkstra, SIGPLAN Notices 13(10):22-26 (Oct 1978).

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

Yellow

keyword: HTML color name



YellowGreen

keyword: HTML color name



Yerk

programming language

Yerk - (named for Yerkes Observatory) A public domain reincarnation of Neon. ftp://oddjob.uchicago.edu/pub/Yerk info: Bob Lowenstein <rfl@oddjob.uchicago.edu>

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

YES

keyword: Objective-C word

yield

keyword: JavaScript reserved word

keyword: PHP reserved word

keyword: Ruby reserved word

YLISP

428

Edition 0.35

abbreviation

programming language

6

YLISP - Hewlett-Packard. A variant of Xlisp for the HP-95LX palmtop. ftp://hpcsos.col.hp.com/mirrors/.scsi5/hp95lx/languages/ylisp*

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

Zz

Z

ASCII: 5A hex

UNICODE: LATIN CAPITAL LETTER Z

Z

ASCII: 7A hex

UNICODE: LATIN SMALL LETTER Z

See: .z — file extension

Z

programming language

year created: 1980

1. ("zed") Programming Research Group, Oxford U, ca 1980. A specification language based on axiomatic set theory and first order predicate logic. Uses many non-ASCII symbols. Used in the IBM CICS project? "Understanding Z", J.M. Spivey, Cambridge U Press 1988. An embedding of Z in HOL available from ICL <ProofPower-server@win.icl.co.uk>

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas Edition 0.35

Z

programming language

2. A stack-based, complex arithmetic simulation language. ZOLA Technologies, ZOLA@Applelink.Apple.com.

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

Z++

programming language

year created: 1991

6

Z++ - Object-oriented extension of Z. "Z++, an Object-Oriented Extension to Z", Lano, Z User Workshop, Oxford 1990, Springer Workshops in Computing, 1991, pp.151-172.

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

See: .zap — file extension

ZAP

programming language

year created: 1982

6

ZAP - Language for expressing transformational developments. "A System for Assisting Program Transformation", M.S. Feather, ACM TOPLAS 4(1):1-20 (Jan 1982).

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

Zed

programming language

year created: 1978

6

Zed - 1978. Software Portability Group, U Waterloo. Eh, with types added. Similar to C. Implementation language for the Thoth realtime operating system. Added a few simple types for greater efficiency on byte-addressed machines. String constants in case statements. Enforces the naming convention: MANIFESTS, Externals and locals. "Porting the Zed Compiler", G.B. Bonkowski et al, SIGPLAN Notices 14(8):92-97 (Aug 1979).

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

ZENO

programming language

year created: 1978

ZENO - U Rochester 1978. Euclid with asynchronous message-passing. "Preliminary ZENO Language Description", J.E. Ball et al, SIGPLAN Notices 14(9):17-34 (Sep 1979).

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas
zero

keyword: COBOL reserved word (z/OS)

ZERO

programming language

year created: 1992

6

ZERO - Object oriented extension of Z. "Object Orientation in Z", S. Stepney et al eds, Springer 1992.

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

zeroes

keyword: COBOL reserved word (z/OS)

zeros

keyword: COBOL reserved word (z/OS)

ZEST

programming language

year created: 1992

ZEST - Object oriented extension of Z. "Object Orientation in Z", S. Stepney et al eds, Springer 1992.

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

magnusoft ZETA

operating system

Proprietary OS devloped by yellowTAB, based on BeOS r5.1d0 source code. —<u>Wikipedia: List of</u> operating systems

ZetaLisp

programming language

year created: 1981

ZetaLisp - Maclisp dialect used on the LISP Machine. The many extensions to Maclisp include vectors, closures, flavors, stack groups, locatives and invisible pointers. "LISP Machine Manual", D. Weinreb and D. Moon, MIT AI Lab, 1981.

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

ZIL

abbreviation

programming language

435

ZIL - Zork Implementation Language. Language used by Infocom's Interactive Fiction adventure games. Interpreted by the zmachine, for Unix and Amiga. ftp://plains.nodak.edu/Minix/st.contrib.Infocom.tar.Z

--The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

See: .zip — file extension

zipcode

programming language

Zipcode - [?] Parallel language at Lawrence Livermore?

---**The Language List:** information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

ZOPL

abbreviation

programming language

ZOPL - Geac. [?] A low-level Pascal?

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

abbreviation

programming language

Sh with list processing and database enhancements. ftp://cs.ucsd.edu/pub/zsh/zsh2.1.0.tar.Z

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

ZUG

programming language

ZUG - Geac. [?] A low-level Awk?

-**The Language List:** information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

Zuse

programming language

year created: 1991

Zuse - (named for Konrad Zuse, the designer of the first modern programming language <u>Plankalkul</u>.) Christian Collberg <collberg@dna.lth.se>, PhD thesis 1991. A descendant of Ada, Modula-2, Mesa and Oberon-1 supporting several levels of information hiding. The Zuse type system includes fully hidden types (similar to Modula-2 opaque types but without any implementation restriction), semi-open pointer types (same as Modula-2 opaque types), extensible record types (similar to Oberon-1 public projection types but without the compiler hint), enumeration types, extensible enumeration types, and extensible subrange types. A type

436

can also be protected by specifying the operations that particular modules may perform (similar to C++ friend classes and Ada private types). Zuse also includes hidden and extensible constants and hidden inline procedures. In order to support the higher levels of information hiding the implementation employs partial intermediate code linking. Version for Sun-3.

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

Zz

programming language

Zz - An extensible language. http://slacvm.slac.stanford.edu:5080/FIND/FREEHEP/NAME/ZZ/FULL

-The Language List: information from The Language List, started 7 March 1991 by Tom Rombouts, then continued by Bill Kinnersley of University of Kansas

dígits

digits 0 1 2 3 4 5 6 7 8 9

zero

0

ASCII: 30 hex

UNICODE: DIGIT ZERO

one

1

ASCII: 31 hex

UNICODE: DIGIT ONE

1410 Processor Operating System (PR-155)

operating system

Proprietary OS by IBM

for the 1410 and 7010 - Wikipedia: List of operating systems



two

2

ASCII: 32 hex

UNICODE: DIGIT TWO

See: .264 55 — file extension

three

3

ASCII: 33 hex

UNICODE: DIGIT THREE

See: .3g2 🥝 — file extension

See: .3ga - file extension

See: .3gp 📿 — file extension



four

4

ASCII: 34 hex

UNICODE: DIGIT FOUR



five

ASCII: 35 hex

UNICODE: DIGIT FIVE



SÍX

6

ASCII: 36 hex

UNICODE: DIGIT SIX

seven

ASCII: 37 hex

UNICODE: DIGIT SEVEN

7040/7044 Operating System (16/32K)

operating system

Proprietary OS by IBM

7040-PR-150 - Wikipedia: List of operating systems

.7z

file extension

Compression: 7-Zip compressed file



eight

ASCII: 38 hex

UNICODE: DIGIT EIGHT



níne

9

ASCII: 39 hex

UNICODE: DIGIT NINE

control codes

NULL
SOH (start of heading)
STX (start of text)
ETX (end of text)
EOT (end of transmission)
SEL (select)
ENQ (enquiry)
ACK (acknowledge)
RNL (require new line)
BELL
BS (backspace)
GE (graphic escape)
HT (horizontal tabulation)
SPS (superscript)
LF (line feed) or NL (new line)
RPT (repeat)

VT (vertical tabulation)
FF (form feed)
CR (carriage return)
SO (shift out)
SI (shift in)
DLE (data link escape)
DC1 (device control 1) or XON (transmit on)
DC2 (device control 2) or TAPE
DC3 (device control 3) or XOFF (transmit off)
DC4 (device control 4) or TAPE
RES/ENP (retore, enable presentataion)
NAK (negative acknowledge)
NL (new line)
SYNC (synchronous idle)
ETB (end of transmission block)
POC (program operator control)
CAN (cancel)
EM (end of medium)
SUB (substitute)
UBS (unit backspace)
ESC (escape)

CU1 (customer use one)

FS (file separator)

GS (group separator)

RS (record separator)

US (unit separator)

IUS/ITB (intermediate transmisson block)

DS (digit select)

SOS (start of significance)

FS (field separator)

WUS (word underscore)

DEL (delete)

null

NUL — NULL

ASCII: 00 hex (originally [1963] called NULL)

EBCDIC: 00 hex

UNICODE: NULL

SOH

start of heading

SOH — Start of Heading

ASCII: 01 hex (originally [1963] called SOM)

EBCDIC: 01 hex

UNICODE: START OF HEADING

STX

start of text

STX — Start of Text

ASCII: 02 hex (originally [1963] called EOA)

EBCDIC: 02 hex

UNICODE: START OF TEXT

ETX — End of Text

ASCII: 03 hex (originally [1963] called EOM)

EBCDIC: 03 hex

UNICODE: END OF TEXT

EOT

end of transmission

EOT — End of Transmission

ASCII: 04 hex

UNICODE: END OF TRANSMISSION

EOT

end of transmission

SEL — Select

EBCDIC: 04 hex

Device control character taking a single-byte parameter

ENQ

enquíry

ENQ — Enquiry

ASCII: 05 hex (originally [1963] called WRU)

UNICODE: ENQUIRY

ACK

acknowledge

ACK — Acknowledge

ASCII: 06 hex (originally [1963] called RU)

UNICODE: ACKNOWLEDGE

460

RNL

require new line

RNL — Require New Line

EBCDIC: 06 hex

Line-break resetting indent Tab mode

BEL bell

BEL — **BELL**

ASCII: 07 hex (originally [1963] called BELL)

UNICODE: BELL

Control-G

BS

backspace

BS—**Backspace**

ASCII: 08 hex (originally [1963] called FE0)

EBCDIC: 16 hex

UNICODE: BACKSPACE

GE graphc escape

GE — Graphic Escape

EBCDIC: 08 hex

Non-locking shift that changes the interpretation of the following character

HT

horizontal tabulation

HT — Horizontal Tabulation

ASCII: 09 hex (originally [1963] called either HT or SK)

EBCDIC: 05 hex

UNICODE: CHARACTER TABULATION

Also known as tab.

SPS

superscript

SPS — Superscript

EBCDIC: 09 hex

Begin superscript or undo subscript.

LF líne feed

LF — Line Feed

ASCII: 0A hex (originally [1963] called VTAB)

UNICODE: LINE FEED

Also known as new line (NL)

Also known as end of line (EOL)

RPT

repeat

RPT — Repeat

EBCDIC: 0A hex

Switch to an operation mode repeating a print buffer.
vertical tabulation

VT — Vertical Tabulation

ASCII: 0B hex

EBCDIC: 0B hex

UNICODE: LINE TABULATION

FF formfeed

FF—**Form Feed**

ASCII: 0C hex

EBCDIC: 0C hex

UNICODE: FORM FEED

CR

carríage return

CR — Carriage Return

ASCII: 0D hex

EBCDIC: 0D hex

UNICODE: CARRIAGE RETURN

SO

shift out

SO — Shift Out

ASCII: 0E hex

EBCDIC: 0E hex

UNICODE: SHIFT OUT

Also knw as LOCKING-SHIFT ONE

51

shift in

SI — Shift In

ASCII: 0F hex

EBCDIC: 0F hex

UNICODE: SHIFT IN

Also known as LOCKING-SHIFT ZERO

DLE

data línk escape

DLE — Data Link Escape

ASCII: 10 hex (originally [1963] called DC0)

EBCDIC: 10 hex

UNICODE: DATA LINK ESCAPE

DC1

device control 1

DC1— Device Control 1

ASCII: 11 hex

EBCDIC: 11 hex

UNICODE: DEVICE CONTROL ONE

Also known as XON (transmit on); used to start paper tape reader

Control-Q

DC2

device control 2

DC2 — Device Control 2

ASCII: 12 hex

EBCDIC: 12 hex

UNICODE: DEVICE CONTROL TWO

Also known as TAPE; used to start paper tape punch

Control-R

DC3

device control 3

DC3 — Device Control 3

ASCII: 13 hex

EBCDIC: 13 hex

UNICODE: DEVICE CONTROL THREE

Also known as XOFF (transmit off); used to stop paper tape reader

Control-S

DC4

device control 4

DC4 — Device Control 4

ASCII: 14 hex

UNICODE: DEVICE CONTROL FOUR

Also known as TAPE; used to stop paper tape punch

Control-T

RES/ENP

restore, enable

presentation

RES/ENP — Restore, Enable, Presentation

EBCDIC: 14 hex

Resume output (after BYP/INP).

NAK

negative

acknowledge

NAK — Negative Acknowledge

ASCII: 15 hex (originally [1963] called ERR)

UNICODE: NEGATIVE ACKNOWLEDGE

NL new líne

NL — New Line

extended ASCII: 85 hex

EBCDIC: 15 hex

Line break.

SYN

sync

SYN — Synchronous Idle

ASCII: 16 hex (originally [1963] called SYNC)

UNICODE: SYNCHRONOUS IDLE

ETB

end of transmission block

ETB — End of Transmission Block

ASCII: 17 hex (originally [1963] called LEM)

UNICODE: END OF TRANSMISSION BLOCK

program operator communícaton

POC — **Program Operator Communicaton**

EBCDIC: 17 hex

Followed by two one-byte operators that identify the specific function, for example a light or function key.

484



cancel

CAN — Cancel

ASCII: 18 hex (originally [1963] called S0)

EBCDIC: 18 hex

UNICODE: CANCEL

EM

end of medium

EM — End of Medium

ASCII: 19 hex (originally [1963] called S1)

EBCDIC: 19 hex

UNICODE: END OF MEDIUM

SUB

substitute

SUB — Substitute

ASCII: 1A hex (originally [1963] called S2; then [1965] called SS)

UNICODE: SUBSTITUTE

UBS

unit backspace

UBS — Unit Backspace

EBCDIC: 1A hex

A fractional backspace.

ESC

escape

ESC — Escape

ASCII: 1B hex (originally [1963] called S3)

UNICODE: ESCAPE

CU1

customer use one

CU1 — Customer Use One

EBCDIC: 1B hex

Not used by IBM; for customer use.

FS file separator

FS — File Separator

ASCII: 1C hex (originally [1963] called S4)

EBCDIC: 1C hex (called IFS)

UNICODE: INFORMATION SEPARATOR FOUR

GS

group separator

GS — Group Separator

ASCII: 1D hex (originally [1963] called S5)

EBCDIC: 1D hex (called IGS)

UNICODE: INFORMATION SEPARATOR THREE

RS

record separator

RS — Record Separator

ASCII: 1E hex (originally [1963] called S6)

EBCDIC: 1E hex (called IRS)

UNICODE: INFORMATION SEPARATOR TWO

US

unit separator

US — Unit Separator

ASCII: 1F hex (originally [1963] called S7)

UNICODE: INFORMATION SEPARATOR ONE

IUS/ITB

intermediate

transmission block

IUS/ITB — Interchange Unit Separator/Intermediate Transmission Block

EBCDIC: 1F hex

Either used as an information separator to terminate a block called a "unit" (as in ASCII; see <u>US</u>), or used as a transmission control code to delimit the end of an intermediate block.

495

DS

dígít select

DS — **Digit Select**

EBCDIC: 20 hex

Used by S/360 CPU edit (ED) instruction.

SOS

start of significance

SOS — Start of Significance

EBCDIC: 21 hex

Used by S/360 CPU edit (ED) instruction.

field separator

FS

FS — Field Separator

EBCDIC: 22 hex

Used by S/360 CPU edit (ED) instruction.

WUS

word underscore

WUS — Word Underscore

EBCDIC: 23 hex

Underscores the immediately preceding word.

DEL

delete

DEL — **Delete**

ASCII: 7F hex

EBCDIC: 07 hex

UNICODE: DELETE

mathematical operators

+ (plus sign)
- (minus sign)
* (multiplication)
/ (division)
< (less than sign)
= (equals sign)
> (greater than sign)

+

ASCII: 2B hex

UNICODE: PLUS SIGN

+

keyword: Ada operator

keyword: BASIC operator

keyword: C operator

keyword: JavaScript operator

keyword: Pascal operator

keyword: PHP operator

keyword: PL/I operator

Milo's Dictionary (with Programming Lexicons)

keyword: Python operator

keyword: Objective-C operator

keyword: Ruby operator

mínus sígn

-

ASCII: 2D hex

UNICODE: HYPHEN-MINUS

Also known as hyphen

-

keyword: Ada operator

keyword: BASIC operator

keyword: C operator

keyword: JavaScript operator

keyword: Pascal operator

keyword: PHP operator
keyword: PL/I operator

keyword: Python operator

keyword: Objective-C operator

keyword: Ruby operator

*

asterísk

*

ASCII: 2A hex

UNICODE: ASTERISK

Also known as star

*

keyword: Ada operator

keyword: BASIC operator

keyword: C operator

keyword: JavaScript operator

keyword: Pascal operator

keyword: PHP operator

keyword: PL/I operator

keyword: Python operator

keyword: Objective-C operator

keyword: Ruby operator

forward slash

/

ASCII: 2F hex

UNICODE: SOLIDUS

Also known as slash

Also known as forward slash

Also known as virgule

/

keyword: Ada operator

keyword: BASIC operator

keyword: C operator

keyword: JavaScript operator

keyword: Pascal operator

- keyword: PL/I operator
- keyword: Python operator
- keyword: Objective-C operator
- keyword: Ruby operator



less than sign

<

ASCII: 3C hex

UNICODE: LESS-THAN SIGN



equals sign

=

ASCII: 3D hex

UNICODE: EQUALS SIGN



greater than sign

>

ASCII: 3E hex

UNICODE: GREATER-THAN SIGN

punctuation and symbols

b (space character)
! (exclamation mark)
! (quotation mark)
(number sign)
\$ (dollar sign)
% (percent sign)
& (ampersand)
' (apostrophe)
((left parenthesis)
) (right parenthesis)
* (asterisk)
+ (plus sign)

Milo's Dictionary (with Programming Lexicons)

514

, (comma)
- (hypen)
. (period)
/ (forward slash)
: (colon)
; (semicolon)
< (less than sign)
= (equals sign)
> (greater than sign)
? (question mark)
(a) (commercial at sign)
[(left square bracket)
\ (backslash)
] (right square bracket)
^ (circumflex accent)
_ (underscore)
` (grave accent)
{ (left curly bracket)
(vertical line)
} (right curly bracket)
~ (tilde)
μ (micro)

ħ

space character

ħ

ASCII: 20 hex

UNICODE: SPACE

]

exclamation mark

!

ASCII: 21 hex

UNICODE: EXCLAMATION MARK

Also known as factorial

11

quotation mark

11

ASCII: 22 hex

UNICODE: QUOTATION MARK

Also known as double quote

#

number sign

#

ASCII: 23 hex

UNICODE: NUMBER SIGN

Also known as pound sign (weight)

Also known as hash tag or hash



dollar sígn

\$

ASCII: 24 hex

UNICODE: DOLLAR SIGN



percent sign

%

ASCII: 25 hex

UNICODE: PERCENT SIGN

See: %ACT

%ACTVATE

keyword: PL/I keyword

Abbreviated: %ACT

Preprocessor statement

%CONTROL

keyword: PL/I keyword

Listing control statement

See: %DCL

See: %DEACT

%DEACTIVATE

keyword: PL/I keyword

Abbreviated: %DEACT

Preprocessor statement

%DECLARE

keyword: PL/I keyword

Abbreviated: %DCL

Preprocesor statement

%DO

keyword: PL/I keyword

Preprocessor statement

%ELSE

keyword: PL/I keyword

Clause of %IF statement

%END

keyword: PL/I keyword

Milo's Dictionary (with Programming Lexicons)

Preprocessor statement

522



ampersand

&

ASCII: 26 hex

UNICODE: AMPERSAND

originally derived from a ligature of 'e' and 't'

&

keyword: C operator

apostrophe

ASCII: 27 hex

۱

UNICODE: APOSTROPHE

Also known as single quote

Also known as APL quote

l left parenthesis

(

ASCII: 28 hex

UNICODE: LEFT PARENTHESIS

Also known as opening parenthesis

right parenthesis

)

ASCII: 29 hex

UNICODE: RIGHT PARENTHESIS

Also known as closing parenthesis

)

comma

,

ASCII: 2C hex

UNICODE: COMMA

hyphen

-

ASCII: 2D hex

UNICODE: HYPHEN-MINUS

Also known as minus sign



ASCII: 2E hex

UNICODE: FULL STOP

Also known as dot

Also known as decimal point

.264

file extension

Video: H.264/MPEG-4 AVC video data file 55

.3g2

file extension

Video: 3GPP2 multimedia data 🥝

.3ga

file extension

Audio: Audio file for wireless networks

.3gp

file extension

Video: Multimedia files for wireless networks

.aac

file extension

Audio: Advanced audio coding stream file 📥

.abr

file extension:

2D graphics: Adobe Photoshop brush

.ai

file extension:

Vector graphics: Adobe Illustrator graphics 📷

.aif

file extension

.aiff

file extension

Audio: Audio Interchange File Format 💿

.amr

file extension

Audio: Adaptive Multi-Rate compressed audio 🥝

.ani

file extension:

2D graphics: Animated cursor 🌌

.ape

file extension

Audio: Monkey's Audio lossless audio compression format ...

.arf

file extension

Audio and Video: WebEx advanced record file

.arj

531

file extension

Compression: ARJ compressed file

.arw

file extension:

Digital Camera RAW photos: Sony Digital Camera RAW digital image

.asf

file extension

Audio and Video: Advanced Systems (streaming) format 💿

.asm

file extension

PC-DOS: assembler source file

.asx

file extension

Audio and Video: Microsoft Advanced Streaming Redirector 💿

.avi

file extension

Video: Audio Video Interleave movie 📓

532

.bak

file extension

PC-DOS: backup files created by EDLIN

.bas

file extension

PC-DOS: BASIC program source file

.bat

file extension

PC-DOS: batch file

.bik

file extension

Video: BINK video 🞇

.bin

file extension

Disc: Binary disc image

PC-DOS: binary image file

.bmp

file extension

Bitmap: Standard Windows bitmap image

.cda

file extension

Audio: CD Audio track file 🕑

.cdr

file extension:

Vector graphics: CorelDRAW vector or bitmap graphics 🕅

.cdt

file extension:

2D graphics: CorelDRAW template

.com

file extension

PC-DOS: command file

.cpt

file extension

Bitmap: Corel PHOTO-PAINT bitmap image

.cr2

file extension:

Digtial Camera RAW photos: Canon digital camera RAW image format version 2.0 💼

.crf

file extension

PC-DOS: cross reference source file

.crw

file extension:

Digtial Camera RAW photos: Canon digital camera RAW image format 🛅

.csh

file extension:

Vector graphics: Adobe Photoshop custom shape 🖪

.csl

file extension:

Vector graphics: CorelDRAW symbol library 🔊

.CSV

file extension

Database: Comma separated value file

.dash

file extension

Video: Dynamic Adaptive Streaming over HTTP video

.dat

file extension

Data: Data file

.dat

file extension

Video: Video CD MPEG movie MPEG1

.db

file extension

Database: Database file

.dbf

file extension

Database: Database file

.dcr

file extension:

Digtial Camera RAW photos: Kodak Digital Camera Raw Image Format ⁵³⁶

.dds

file extension

Bitmap: DirectDraw surface data

.deb

file extension

Compression: Debian software package file

.dib

file extension

Bitmap: Device-Independent bitmap graphics 🔤

.djvu

file extension:

2D graphics: DjVu image 🎴

.dmg

file extension

Disc: macOS X disk image

.dng

Milo's Dictionary (with Programming Lexicons)

file extension

Bitmap and Digtial Camera RAW photos: Adobe Digital Negative

.drw

file extension:

Vector graphics: Vector drawing

.dvf

file extension

Audio: Sony DV voice data

.dvr

file extension

Video: Microsoft Windows XP Media Center TV recording 🌌

.email

file extension

E-mail: Outlook Express e-mail message file

.emf

file extension

Bitmap: Enhanced Windows Metafile picture 🚨

.eml

file extension

E-mail: e-mail message file from mulitple e-mail clients

.emix

file extension

E-mail: Apple Mail e-mail file

.emz

file extension:

Vector graphics: Microsoft Windows compressed enhanced metafile

.eps

file extension:

2D graphics: Encapsulated PostScript image

.exe

file extension

PC-DOS: executable file, created by linker

.fla

file extension:

2D graphics: Editable Adobe Flash project (animation, movie, flash application)

.flac

file extension

Audio: Audio files encoded by Flac - free lossless audio codec

.flv

file extension

Video: Flash video 💋

.fpx

file extension:

Digtial Camera RAW photos: Kodak FlashPiX bitmap image 📓

.gif

file extension

Bitmap: Graphics interchange file format 🌺

.gp

file extension

Audio: Guitar Pro 7 project

.gp3

file extension 540
.gp4

file extension

Audio: Guitar Pro 4 project 😹

.gp5

file extension

Audio: Guitar Pro 5 project 😹

.gpt

file extension

Audio: Guitar Pro 1 project

.gpx

file extension

Audio: Guitar Pro 6 project

.h264

file extension

Video: H.264 video 🥝

.heic

541

file extension

Bitmap: High Efficiency Image Format

.hex

file extension

PC-DOS: hex ASCII file which DEBUG converts to binary and loads

.HTM

file extension

Websites: HyperText Markup Language

.HTML

file extension

Websites: HyperText Markup Language

.icns

file extension:

2D graphics: Apple icon graphic format

.ico

file extension

Bitmap: Icon image 📽

.icon

file extension

Bitmap: Icon image file 馨

.iso

file extension

Disc: ISO disc image

.jpeg

file extension

Image: Joint Photographic Experts Group bitmap image

.jpg

file extension

Image: Joint Photographic Experts Group bitmap image 🖻

.lib

file extension

PC-DOS: source code library file

.logic

file extension

.lst

file extension

PC-DOS: listing file made by assembler

.m2t

file extension

Video: MPEG-2 transport stream



.m2ts

file extension

Video: MPEG-2 stream (Blu-Ray)

.m4a

file extension

Audio: MPEG-4 compressed audio 💿

.m4b

file extension

Audio: MPEG-4 Audio Layer audio book 🥝

.m4p

file extension

Audio: Apple iTunes Music Store audio @

.m4v

file extension

Video: MPEG-4 video file format 🛍

.map

file extension

PC-DOS: linker's map

.mdb

file extension

Database: Microsoft Access database file

.mdi

file extension:

2D graphics: Microsoft Office Document Imaging file format

.mid

file extension

Audio: MIDI audio file

.midi

file extension

Audio: MIDI audio file

.mkv

file extension

Video: Matroska multimedia container video file 🛍

.mov

file extension

Video: Apple QuickTime digital video 🥝

.mp3

file extension

Audio: MP3 compressed audio file (audio, music, or rigntones)

.mp4

file extension

Video: MPEG-4 video file format 🥝

.mpa

file extension 546

.mpeg

file extension

Video: MPEG movie 🗿

.mpg

file extension

Video: MPEG 1 video file format 🗐

.mrw

file extension:

Digtial Camera RAW photos: Minolta Dimage RAW image 💼

.msg

file extension

E-mail: Microsoft Outlook e-mail message file

.mts

file extension

Video: AVCHD MPEG-2 transport stream

.nef

file extension:

Digtial Camera RAW photos: Nikon Digital SLR camera RAW image 🛅

.obj

file extension

PC-DOS: assembled object file

.odg

file extension:

2D graphics: OpenDocument drawing 🚺

.ods

file extension

Spreadsheet: Openoffice spreadsheet file

.odt

file extension

Word Processing: Openoffice text document

.oft

file extension

E-mail: Microsoft Outlook e-mail template file

.0gg

file extension

Audio: Ogg Vorbis audio file 💿

.0gv

file extension



.opus

file extension

Audio: Opus audio data

.orf

file extension:

Digtial Camera RAW photos: Olympus Digital camera raw image 🕅

.ost

file extension

E-mail: Microsoft Outlook offline e-mail storage file

.pcd

Milo's Dictionary (with Programming Lexicons)

file extension:

Digtial Camera RAW photos: Kodak Picture CD multiresolution image 🤯

.pcm

file extension

Audio: PCM audio data file

.pcx

file extension

Bitmap: Paintbrush bitmap image 💱

.pdf

file extension

Text: Adobe Portable Document File

.pic

file extension

Bitmap and Vector graphics: Picture image

.pkg

file extension

Compression: Package file

.png

file extension

Bitmap: Portable Network Graphic file

.ppt

file extension

Presentation: Microsoft PowerPoint file

.pptx

file extension

Presentation: Microsoft PowerPoint file (more recent than the older .ppt)

.prproj

file extension

Video: Adobe Premiere Pro project 📴

.psb

file extension:

2D graphics: Adobe Photoshop Large Document Format 📧

.psd

file extension

Ps

.psdx

file extension

Bitmap: Adobe Photoshop Touch graphics

.pst

file extension

E-mail: Microsoft Outlook e-mail storage file

.ptx

file extension:

Digtial Camera RAW photos: Pentax RAW bitmap graphic file

.pzl

file extension:

2D graphics: Puzzle picture image 🤯

.raf

file extension:

Digtial Camera RAW photos: Fuji CCD-RAW graphic image 🛅

.rar

file extension

Compression: RAR file

.raw

file extension

Bitmap and Digtial Camera RAW photos: Digital camera photo RAW image format

.rec

file extension

Audio and Video: Recorded data

.ref

file extension

PC-DOS: cross reference listing file

.rel

file extension

PC-DOS: relocatable file from assembler

.rmvb

file extension

Video: RealMedia variable bitrate 🧐

.rpm

file extension

Compression: red Hat Package Manager

.rw2

file extension:

Digtial Camera RAW photos: Panasonic Lumix RAW image

.sav

file extension

Data: Save file (especially a game save file)

.sda

file extension:

Vector graphics: Apache OpenOffice (OpenOffice.org) drawing

.snd

file extension

Audio: Sound data

.sng

file extension

Audio: song file

554

.sql

file extension

Database: SQL database file

.sup

file extension:

3D graphics: 3D Subtitle format

.svg

file extension:

Vector graphics: XML based scalable vector graphics format 🍒

.svgz

file extension:

Vector graphics: Compressed SVG (Scalable Vector Graphics)

.swf

file extension

Video and Vector graphics: ShockWave Flash, Animated vector format for the Internet 💋

.tar

file extension

555

Dat: Unix or Linux tarball file archive

.tar.gz

file extension

Compression: Tarball compressed file

.tga

file extension

Bitmap: Truevision TarGA bitmap image

.thm

file extension

Bitmap: Thumbnail bitmap image

.tif

file extension

Bitmap: Aldus Tagged Image File Format (TIFF) bitmap image

.tiff

file extension

Bitmap: Aldus Tagged Image File Format (TIFF) bitmap image

.toast

file extension

Disc: Toast disc image

.tod

file extension

Video: Hard disk camera movie MPEG-2 transport stream



.tmp

file extension

PC-DOS: temporary file

.tp

file extension

Video: MPEG-2 TV recorded video

.ts

file extension

Video: MPEG-TV recorded video format

.txt

file extension

Text: plain text file (normally ASCII-only)

.uax

file extension

Audio: Unreal audio package 🥨

.vcd

file extension

Disc: Virtual CD

.vcf

file extension

E-mail: e-mail contact file

.vob

file extension

Video: DVD video object 📟

.vsdx

file extension:

2D graphics: Microsoft Visio drawing

.wav

558

file extension

Audio: WAVe PCM Sound, the standard Windows sound format 🗐

.wbmp

file extension

Bitmap: Wireless Bitmap image

.wdp

file extension

Bitmap: Windows Media HD Photo picture image

.webm

file extension

Video: Matroska WebM video 🚺

.webp

file extension

Bitmap: Google image file format

.wimp

file extension

Video: Microsoft Windows Live Movie Maker video project 🌌

.wma

file extension

Audio: Windows Media Audio file 🗿



.wmv

file extension

Video: Windows Media Video 🗿

.wpl

file extension

Audio: Windows Media Player playlist

.xls

file extension

Numerics: Microsoft Excel file

.xlsx

file extension

Spreadsheet: Microsoft Excel file (more modern than the older .xls)

.xml

560

file extension

Data: XML file

.xmp

file extension:

2D graphics: Adobe Extensible Metadata Platform data

.Z

file extension

Compression: Z compressed file

.zap

file extension

Audio: Zipped audio book

.zip

file extension

Compression: Zip compressed file

.\$\$\$

file extension

PC-DOS: EDLIN or other temporary file

:

ASCII: 3A hex

562

UNICODE: COLON

◆◆

colon

;

semícolon

;

ASCII: 3B hex

UNICODE: SEMICOLON

question mark

?

ASCII: 3F hex

UNICODE: QUESTION MARK



at sígn

a

ASCII: 40 hex

UNICODE: COMMERICAL AT

Also known as the at sign

@catch()

keyword: Objective-C word

*a*class

keyword: Objective-C word

@dynamic

keyword: Objective-C word 565

@end

keyword: Objective-C word

*a*finally

keyword: Objective-C word

*@*implementation

keyword: Objective-C word

*a*interface

keyword: Objective-C word

*a*private

keyword: Objective-C word

@property

keyword: Objective-C word

@protected

keyword: Objective-C word

@protocol

keyword: Objective-C word

@public

keyword: Objective-C word

@select

keyword: Objective-C word

asynthesize

keyword: Objective-C word

*a*throw

keyword: Objective-C word

atry

keyword: Objective-C word

l left square bracket

[

ASCII: 5B hex

UNICODE: LEFT SQUARE BRACKET

Also known as opening square bracket

backslash

\

ASCII: 5C hex

UNICODE: REVERSE SOLIDUS

Also known as escape character

l ríght square bracket

]

ASCII: 5D hex

UNICODE: RIGHT SQUARE BRACKET

Also known as closing square bracket



círcumflex accent

۸

ASCII: 5E hex (orginally [1963] ↑)

UNICODE: CIRCUMFLEX ACCENT

underscore

ASCII: 5F hex (originally [1963] \leftarrow)

UNICODE: LOW LINE

_Bool

keyword: Objective-C reserved word

CLASS

keyword: PHP reserved word

_Complex

keyword: Objective-C reserved word 572

DIR

keyword: PHP reserved word

ENCODING

keyword: _ENCODING_ (all caps) Ruby reserved word

FILE

keyword: PHP compile-time constant

keyword: _FILE_ (all caps) Ruby reserved word

FUNCTION

keyword: PHP compile-time constant

halt_compiler()

keyword: PHP reserved word

_Imaginary

keyword: Objective-C reserved word

LINE

keyword: PHP compile-time constant

keyword: _LINE_ (all caps) Ruby reserved word

METHOD

keyword: PHP compile-time constant

NAMESPACE

keyword: PHP compile-time constant

TRAIT_

keyword: PHP compile-time constant

grave accent

ASCII: 60 hex

٦

UNICODE: GRAVE ACCENT

Also known as backtick

left curly bracket

{

ASCII: 7B hex

UNICODE: LEFT CURLY BRACKET

Also known as open curly bracket

Also known as left brace
l vertical line

ASCII: 7C hex

UNICODE: VERTICAL LINE (originally [1963] ACK [acknowledge], then [1965] ¬)

Also known as vertical bar

Also known as pipe

ríght curly bracket

}

ASCII: 7D hex (originally [1963] ESC [escape], then [1965] |)

UNICODE: RIGHT CURLY BRACKET

Also known as closing curly bracket

Also known as right brace

tilde

 \sim

ASCII: 7E hex

UNICODE: TILDE

μ mícro

μ

UNICODE: 00B5 MICRO SIGN

μ-velOSity

operating system

Proprietary OS by Green Hills Software

Greek alpha A α
Greek beta B β
gamma Γ γ
delta Δ δ
epsilon E ε
zeta Z ζ
eta Η η
theta Θθ ϑ
iota I ı
kappa К к
lam(b)da Λ λ
mu M μ

Milo's Dictionary (with Programming Lexicons)

nu N v

xiΞξ

omicron O o

рі П **π** ज

rho P p

sigma $\Sigma \sigma \varsigma$

tau T τ

upsilon Y υ Υ

phi **Φ** φ

chi X χ

psi Ψ ψ

omega $\Omega\,\omega$



Alpha A

Greek letter

UNICODE: 0391 GREEK CAPITAL LETTER ALPHA

apha α

Greek letter

UNICODE: 03B1 GREEK SMALL LETTER ALPHA

beta Bβ

Beta B

Greek letter

UNICODE: 0392 GREEK CAPITAL LETTER BETA

beta β

Greek letter

UNICODE: 03B2 GREEK SMALL LETTER BETA



Gamma **F**

Greek letter

UNICODE: 0393 GREEK CAPITAL LETTER GAMMA

gamma y

Greek letter

UNICODE: 03B3 GREEK SMALL LETTER GAMMA



Delta **A**

Greek letter

UNICODE: 0394 GREEK CAPITAL LETTER DELTA

delta δ

Greek letter

UNICODE: 03B4 GREEK SMALL LETTER DELTA

Epsilon E

Greek letter

UNICODE: 0395 GREEK CAPITAL LETTER EPSILON

epsilon ε

Greek letter

UNICODE: 03B5 GREEK SMALL LETTER EPSILON

Ζζ

Zeta Z

Greek letter

UNICODE: 0396 GREEK CAPITAL LETTER ZETA

zeta ζ

Greek letter

UNICODE: 03B6 GREEK SMALL LETTER ZETA

eta Ηη

Eta H

Greek letter

UNICODE: 0397 GREEK CAPITAL LETTER ETA

eta η

Greek letter

UNICODE: 03B7 GREEK SMALL LETTER ETA

theta Θθϑ

Theta **O**

Greek letter

UNICODE: 0398 GREEK CAPITAL LETTER THETA

theta θ

Greek letter

UNICODE: 03B8 GREEK SMALL LETTER THETA

symbol theta v

Greek letter

UNICODE: 03D1 GREEK THETA SYMBOL

Edition 0.35

íota I ı

Iota I

Greek letter

UNICODE: 0399 GREEK CAPITAL LETTER IOTA

iota ı

Greek letter

UNICODE: 03B9 GREEK SMALL LETTER IOTA



Kappa K

Greek letter

UNICODE: 039A GREEK CAPITAL LETTER KAPPA

карра к

Greek letter

UNICODE: 03BA GREEK SMALL LETTER KAPPA

lam(b)da $\Lambda\lambda$

Lambda Λ

Greek letter

UNICODE: 039B GREEK CAPITAL LETTER LAMDA

lambda λ

Greek letter

UNICODE: 03BB GREEK SMALL LETTER LAMDA

mu M _µ

Mu M

Greek letter

UNICODE: 039C GREEK CAPITAL LETTER MU

mu µ

Greek letter

UNICODE: 03BC GREEK SMALL LETTER MU

nu Nv

Nu N

Greek letter

UNICODE: 039D GREEK CAPITAL LETTER NU

nu v

Greek letter

UNICODE: 03BD GREEK SMALL LETTER NU

XÍ Ξξ

Xi Ξ

Greek letter

UNICODE: 039E GREEK CAPITAL LETTER XI

xi ξ

Greek letter

UNICODE: 03BE GREEK SMALL LETTER XI

omícron O o

Omicron O

Greek letter

UNICODE: 039F GREEK CAPITAL LETTER OMICRON

omicron o

Greek letter

UNICODE: 03BF GREEK SMALL LETTER OMICRON

pí Π_πω

Pi Π

Greek letter

UNICODE: 03A0 GREEK CAPITAL LETTER PI

ρί π

Greek letter

UNICODE: 03C0 GREEK SMALL LETTER PI

symbol pi 👦

Greek letter

UNICODE: asdf

600

Edition 0.35

Greek alphabet

600

rho Pp

Rho P

Greek letter

UNICODE: 03A1 GREEK CAPITAL LETTER RHO

rho ρ

Greek letter

UNICODE: 03C1 GREEK SMALL LETTER RHO



Sigma Σ

Greek letter

UNICODE: 03A3 GREEK CAPITAL LETTER SIGMA

sigma σ

Greek letter

UNICODE: 03C3 GREEK SMALL LETTER SIGMA

final sigma ς

Greek letter

UNICODE: 03C2 GREEK SMALL LETTER FINAL SIGMA

Edition 0.35

tau Tτ

Tau T

Greek letter

UNICODE: 03A4 GREEK CAPITAL LETTER TAU

tau τ

Greek letter

UNICODE: 03C4 GREEK SMALL LETTER TAU

upsílon YvY

Upsilon Y

Greek letter

UNICODE: 03A5 GREEK CAPITAL LETTER UPSILON

upsilon υ

Greek letter

UNICODE: 03C5 GREEK SMALL LETTER UPSILON

symbol upsilon Y

Greek letter

UNICODE: 03D2 GREEK UPSILON WITH HOOK SYMBOL

Greek alphabet

)

Edition 0.35

phí Φφ

Phi Φ

Greek letter

UNICODE: 03A6 GREEK CAPITAL LETTER PHI

phi φ

Greek letter

UNICODE: 03C6 GREEK SMALL LETTER PHI



Chi X

Greek letter

UNICODE: 03A7 GREEK CAPITAL LETTER CHI

$chi\,\chi$

Greek letter

UNICODE: 03C7 GREEK SMALL LETTER CHI

psí Ψψ

Psi Ψ

Greek letter

UNICODE: 03A8 GREEK CAPITAL LETTER PSI

psi y

Greek letter

UNICODE: 03C8 GREEK SMALL LETTER PSI



Ωω

Omega Ω

Greek letter

UNICODE: 03A9 GREEK CAPITAL LETTER OMEGA

omega w

Greek letter

UNICODE: 03C9 GREEK SMALL LETTER OMEGA

time line

pre-1950

date	name/event	person/creator/discoverer/inventor	notes
1804	<u>Jacquard</u> machine	Joseph Marie Jacquard	programming language
27 September 1919	born	James Wilkinson	English scientist, mathematician
26 September 1924	born	Jean Hoerni	silicon transistor pioneer
27 September 1930	born	Alan Shugart	pioneer in computer disk drive and founder of Seagate Technology
1943-45	<u>Plankalkül</u>	Konrad Zuse	programming language
1943-46	ENIAC coding	John von Neumann, John Mauchly, J. Presper Eckert, Herman Goldstine after Alan Turing. The first programmers of ENIAC were Kay	programming

	system	McNulty, Betty Jennings, Betty Snyder, Marlyn Meltzer, Fran Bilas, and Ruth Lichterman.	language
1946	ENIAC Short Code	Richard Clippinger, John von Neumann after Alan Turing	programming language
1946	Von Neumann and Goldstine graphing system (Notation)	John von Neumann and Herman Goldstine	programming language
1947	ARC Assembly	Kathleen Booth	programming language
1948	CPC Coding scheme	Howard H. Aiken	programming language
1948	<u>Curry</u> notation system	Haskell Curry	programming language
1948	<u>Plankalkül</u> (year of publication)	Konrad Zuse	programming language
1949	EDSAC Initial Orders	David Wheeler	programming language
1949	<u>Brief Code</u>	John Mauchly and William F. Schmitt	programming language

1950s

date	name/event	person/creator/discoverer/inventor	notes
27 September	born	Larry Wall	created
612	I		Edition 0.35
1954

Perl

1980s

date	name/event	person/creator/discoverer/inventor	notes
27			GNU
September	GNU project	Richard Stallman	project
1983			announced

1980s

date	name/event	person/creator/discoverer/inventor	notes
27 September 1999	Intel Pentium III 533B and 600B	Intel	product release

2010s

date	name/event	person/creator/discoverer/inventor	notes
			Yahoo
			confirmed
			that in 2014
			that the
			account
26			information
September		Yahoo	of around
2016			500 million
			users was
			stolen by
			hackers in a
			network
			breach.

27 September 2010 Cloudflare 2010 Cloudflare Cloudflare Cloudflare Cloudflare Cloudflare Cloudflare	Milo's Dictionary (with	Programming Lexicons)	614
	27 September 2010	Cloudflare	launched its web protection and optimization services

what if language

This section is a historical what if.

overview of thought experiment

What if we could have designed a high quality programming language that was released in the early days of computing and that with time-appropriate extensions stood the test of time and remained useful to the present? While some early languages (such as FORTRAN, BASIC, COBOL, and PL/I) still have some use in maintaining legacy systems, only LISP and FORTRAN remain strong languages in their respective fields (LISP primarily in AI and FORTRAN primarily in scientific use of super computers). And FORTRAN sucks — which you would know if you ever wrote production software in it.

FORTRAN was accepted and was the overwhelming dominant language for more than a decade for one and only one reason: it produced compiled code nearly as efficient as hand assembly.

The creators of FORTRAN had seen well designed languages fail because they were too slow for extremely expensive primitive early computers. They knew that compiled languages would only be accepted if they produced fast object code.

Therefore they put all of their efforts into optimizing for efficient code. Many of their ideas are still in use and the efficiency of FORTRAN I far exceeds any modern compiler.

Unfortunately they ignored language design and FORTRAN was a nightmare for coding. I wrote professional FORTRAN softare. Anyone who wrote FORTRAN code can verify this.

LISP was a great language, but a surprising number of people rejected it because they didn't like counting parenthesis.

COBOL sufered from being way too wordy and for emphasizing features of operating systems that disappeared into history. The creators of COBOL honestly thought that they were creating a programming language that any business person would be able to write code with. After a few years of reality, that goal was changed to making programs that any business person could read and verify that the program accurately relected busines sneeds. Even that goal wasn't true.

615

BASIC was relatively easy to use and lasted a surprisingly long time, both as a hobbyist language and surprisingly sometimes as a professional language. BASIC was cleaner than FORTRAN, but suffered from some of the drawbacks of FORTRAN.

PL/I was an attempt to combine both FORTRAN and COBOL as well as include some ideas that had been developed later, such as pointers and stacks. It had some new problems of its own, but its biggest problem was it was perceived as an IBM-only languages.

APL was a very efficient and terse language. Its big problem was it required a special keyboard. It was briefly popualr two different times because it allowed those with a strong mathematics background to quickly write one-off programs to solve some mathematical problem.

C is an extremely influential langauge. Most modern languages have C as one of their major influences, directly or indirectly through other languages. C did (and still does) have some key problems, especially for implementation on computers where the natural word size was not a multiple of the natural character size and for processors of less than eight bits.

C++, Objective C, and other object oriented extensions of C show how messy things can get when trying to cram new ideas into an established language.

In our thought experiment, we somehow can go back in time and introduce a programming language that is orderly and can over time easily and comfortably accept new ideas.

Our thought experiment is limited by the concern of the original FORTRAN team: each version of our proposed language has to be efficiently implemented on the hardware of the time.

We can have the plans for the series of versions mapped out in advance.

LISP was originally proposed in a 1960 paper by John McCarthy. McCarthy had been experimenting with ideas for LISP in 1958. To McCarthy's surprise, Steve Russell implemented the LISP eval function in IBM 704 machine code.

Our thought experiment is: how would we map out the evolution of one single coherent programming langauge in a manner where each version can efficiently be run on the hardware for each version's own time period.

The rules to this thought experiment:

(1) We can introduce parts of the language over time. We are allowed to take into account how computing changes over time and make sure from the beginning that our language will continue to make sense at each step of the way and that new parts will fit smoothly with old parts.

(2) We can't introduce anything until the technology reasonably supports the things we introduce. This will be the most difficult in the first two to four decades of computing (depending on when you view the first version as coming into existence).

(3) We can't throw away any parts once they are introduced. The entire language must remain 616 Edition 0.35

backwards compatible.

overview of thought experiment
history
source code
identifiers
keywords
literals
constants
operators and separators
white space
new lines
comments
preprocessor
declarations
types
symbols
logical types
numeric types
integer types
real types
complex types
character types

string types	
relational operators and ordered types	
pictures and formats	
pointer types	
reference and dereference	
access types	
dynamic storage	
vector types	
matrix types	
array types	
table types	
ranges	
enumeration types	
tuples	
sets	
structure types	
uniontypes	
record types with discriminants	
hashes	
lists	
stacks	
queues	

dictionaries

objects
function types
void types
user defined types
files
scope and visibility
conversions
expressions
primary expressions
postfix expressions
unary expressions
binary operator expressions
logical operator expressions
logic assertions
conditional expressions
assignment expressions
sequential expressions
constrant expressions
order of evaluation
statements
expression statements
arithmetic statements

IVIIIO 3	s Dictionary (with Frogramming Lexicons)	020
	labeled statements	
	blocks	
	compounds tatements	
	conditional statements	
	loops	
	interative statements and iterators	
	case statements	
	break and continue statements	
	return statements	
	goto statements	
	low-level statements	
	null statements	
	conditions	
	exceptions	
	errors	
	procedures	
	functions	
	recursion	
	methods	
	lambdas	
	input statements	
	output statements	

database statements

classes
modules
packages
generic units
reflection
operations
resources
globals
concurrent execution
tasks
semaphores
asycnhronous message passing
remote procedure call
rendvous
virtualmachines
low-level programming
standard library
standard language additions
character processing functions
string processing functions
memory functions
input output functions

time and date functions

control functions

miscellaneous functions

hístory

a quick history

For those who didn't live through the early days of computing, let me provide you with one example of the severity of constraints of early computing:

In the early 1960s 1401 FORTRAN (which came between the unreleased FORTRAN III and FORTRAN IV) for the IBM 1401 computer used a 63-pass compiler. This complier required 8000 character of core memory (six bits each). It could be run from magnetic tape or from a 2200-card deck. Once started, it used no external memory (other than reading in the code for the next passes of the compiler from tape, cards, or disk storage). The compiler was loaded in a series of overlays (software running consecutively in the same memory locations) and gradually transformed the data in memory.

At the time not all computers had 8000 memory bytes as an option and for those that did have that much or memory, it was very expensive. The magnetic cores had to be hand sewn by (mostly) women working under powerful magnification.

The FORTRAN II compiler had 29 passes, requiring 4K of memory and additional removable cartridge disks totalling 5 megs.

Donald Knuth's personal notes, now digitized and available online, are filled with discussions by early compiler writers (well into the 1960s) about how they had to severely limit the number of language keywords to leave room for also storing variables from prorgams being compiled.

LISP couldn't reasonably be used (other than for John McCarthy's experiments) for a couple of years until someone figured out a clever scheme that made efficient evaluation of LISP possible.

I hope this gives you some idea of the severe limitations for the first releases of this what if language.

early assemblers

Maurice Wilkes and W. Renwick created an assembler called Initial Orders that ran on the EDSAC (Electronic Delay Storage Automatic Calculator) at Cambridge University.

source code

input

Source code can come from files, cards, tapes (paper or magnetic), or some kind of keyboard or terminal.

When the source code is input from files, cards, tapes (paper or magnetic), or some other storage, the program can be either interpreted or compiled.

When the source code is input from some kind of keyboard or terminal, it is an interactive session and each line is immediately interpreted unless the interactive session is in an editing mode.

source code representation

Source code is Unicode text encoded in UTF-8. The text is not canonicalized, so a single accented code point is distinct from the same character constructed from combining an accent and a letter; those are treated as two code points. For simplicity, this document will use the unqualified term character to refer to a Unicode code point in the source text. — *The Go Programming Language Specification*

Each code point is distinct; for instance, upper and lower case letters are different characters. — *The Go Programming Language Specification*

Implementation restriction: For compatibility with other tools, a compiler may disallow the NUL character (U+0000) in the source text. — <u>*The Go Programming Language Specification*</u>

Implementation restriction: For compatibility with other tools, a compiler may ignore a UTF-8encoded byte order mark (U+FEFF) if it is the first Unicode code point in the source text. A byte order mark may be disallowed anywhere else in the source. — <u>The Go Programming Language</u> <u>Specification</u>

The UTF-8 standard recommends replacing each error (which for this language will include the UTF-8-encoded byte order mark (U+FEFF) (other than as the first Unicode code point in the source

text) with the replacement character " \clubsuit " (U+FFDD). When in edit mode, it will be represented by the replacement character, but the original will be preserved unless changed by the user, and the user can examine to see what the underlying data is. Any UTF-8 errors that result in the replacement character " \bigstar " (U+FFDD) are ignored, If the remaining text is valid, the program will continue to be interpreted or compiled. — *The Go Programming Language Specification*

source code format

Portions of this document are provided here. Use the link to view and read the entire document.

NOTE: This document was released almost a year *before* the language was released.

From the IBM Programmer's Reference Manual FORTRAN Automatic Coding System for the IBM 704:

Keypunching the Program

Each statement is punched on a separate card. If a statement is too long to fit on a single card it can be continued over as many as 9 additional *continuation* cards. For each statement the initial card must contain

either a zero or blank in column 6; on continuation cards column 6 must not contain a zero or blank, and it should be used to number the continuation cards consecutively from 1 to 9.

If a statment is too long to fit on a single line of the coding form, the programmer can signal to the keypuncher that he has continued on to the next line by placing a mark in the column labeled CONTINUATION.

The order of the statements is governed solely by the order of the cards. However, any number less than 2^{15} (=32768) may be associated with any statement by punching it in columns 1-5 of the initial card bearing that statemet. Thereupon this number becomes the *statement number* of that statement. Statement numbers, besides permitting cross-references within the source code, also help the programmer to correlate the object program with his source code.

Punching the character C in column 1 will cause the card to be ignored by FORTRAN. Such cards may therefore be used to carry comments which will appear when the deck is listed.

Columns 73-80 are not read by FORTRAN and may be punched with any desired identifying information.

The statements themselves are punched in columns 7-72, both on initial and continuation cards. Thus a statement consists of not more than $10 \ge 660$ characters. A table of the admissible characters in FORTRAN is given in Appendix A.



Blank characters, except in column 6, are simply ignored by FORTRAN, and the programmer may use blanks freely to improve the readability of his FORTRAN listing.

program structure

A PL/I program consists of one or more blocks of statements called procedures. A procedure may be thought of as a subroutine. Procedures may invoke other procedures, and these procedures or subroutines may either be compiled separately, or may be nested within the calling procedure, and compiled with it. Each procedure may contain declarations that define names and control the allocation of storage. — *IBM System/360 Operating System PL/I (F) Language Reference Manual*

identifiers

Identifiers are sequences of letters, digits, and underscores. The first character must be a letter. Case is important; e.g, CS, CS, cS, and CS are all different identifiers. — *The SR Programming Language*

keywords

Keywords are special identifiers, such as if and do, whose meaning is fixed in the language. — *The SR Programming Language*

literals

Literals are specific values of different types, such as booleans, integers, reals, characters, and strings. — *The SR Programming Language*

11 4096 1333Q 0abcdefX

constants

constants

constants

operators and separators

Operators and separators are keywords or special characters, such as <= or []. — *The SR Programming Language*

The integer operators include arithmetic operators, such as +, **, % (remainder), and mod; bitwise operators, such as and and or; and bit-shifting operators, such as >> (right shift). Each operator returns an integer result. — *The SR Programming Language*

white space

White space — blanks or tabs — may appear between any two tokens. It is ignored unless it is essential to separate two tokens, such as a keyword followed by an identifer. — *The SR Programming Language*

new línes

Newline character, too, can sometimes be used as white space. However, they can also indicate the end of a declaration, statement, or expression. When a newline character is encountered, its rule is determined by following rule: If the previous token can legally terminate a declaration, estatement, or expression — and if the following token can legally begin one — then the newline acts as a separator; otherwise it is ignored. For example, consider the following program fragment: *— The SR Programming Language*

It consists of a single assignment, assigning 13 to X. In contrast consider: — *The SR Programming Language*

This consists of two expressions: the first assigns 10 to X, and the second just evaluates the expression + 3. To rewrite the original example with the + on the second line, an explicit line continuation, denoted by λ , is placed at the end of the lie to be continued, as in — *The SR Programming Language*

In addition to newline characters, semicolons can be used to terminate declarations, statements, or expressions. Semicolons also appear in the SR grammar in a few places, such as between field definitions in a record. Semicolons can *always* be replaced by newline characters. — *The SR Programming Language*

comments

SR provides two different kinds of comments. In Chapter 1, we used a one-line comment, which begins with # and ends at the next newline character (or the end of the file). The second kind of comment is a bracketed comment, which begins with /* and ends with the corresponding occurence of */. Bracketed comments can appear within a line, or they can span more than one line. Bracketed comments can also be nested; otherwise all characters within a comment are ignored until the end of the comment is reached. For example, # is ignored within a bracketed comment. — *The SR Programming Language*

preprocessor

preprocessor

preprocessor

declarations

declarations

declarations



types

types

SR has five basic types: boolean, integer, real, character, and string. They are represented by keywords bool, int, real, char, and string, respectively, — *The SR Programming Language*

symbols

symbols

symbols

logícal types boolean types

logical types or boolean types

The boolean type has two literals: false and true. The boolean operators are and, or, xor (exclusive or), and not. Each operator returns a boolean result. — *The SR Programming Language*

Evaluation of boolean expression is *short-circuit*: evaluation stops as sooon as the final value of the expression can be determined. In particualr, evaluations of **and** and **or** are short-circuit. For example, if **and**'s first operand evaluates to false, its second operand is not evaluated because the entire expression is known to be false. Similarly, if **or**'s first operand evaluates to true, its second operand is not evaluated because the entire expression is known to be entire expression is known to be true. *— The SR Programming Language*

asdf

numeríc types

numeric types

numeric types

integer types

11 4096 1333Q 0abcdefX

The integer operators include arithmetic operators, such as +, **, % (remainder), and mod; bitwise operators, such as and and or; and bit-shifting operators, such as >> (right shift). Each operator returns an integer result. — *The SR Programming Language*

The modulo and remainder operators are related in that they both return a result that represents the remainder from the division of their two operands. However, they differ in exactly what they return. The sign of a mod b is that of b, while the sign of a % b depends on how integer division is performed on the underlying machine. For example, consider a circular buffer of size n with slots numbered 0, 1, ..., n-1. The two slots adjacent to slot i have the indices (i + 1) mod n and (i - j) mod n. When i is zero, the value of the last expression is n - 1, whereas the value of (i - j) % n can be either n - 1 or -1. Accordingly, the modulo operator is generally more useful than the remainder operator. — *The SR Programming Language*

A number of predefined fucntions—such as **abs** and **max**—can be applied to integers. — *The SR Programming Language*

real types

A real (floating point) literal has the general form — The SR Programming Language

integer_part . fraction_part exponent_part

Programming Language

11.23.14159270.00..01.23e-431.23E-45.123e-44421e+3421e3

The real operators are the same arithmetic operators as for integers. Each operator returns a real result. The comments regarding the modulo and remainder operators on integers also apply to their real counterparts. — *The SR Programming Language*

As with integers, a number of predefined functions—such as **abs** and **max**—can be applied to reals. Other predefined functions for reals include standard mathematical functions, such as **sin**, **log**, and **sqrt**.s — *The SR Programming Language*

complex types

complex types

complex types

character types

character types

Character values are represented using type **char**. Character literals are single ASCII characters enclosed in single quotes (apostrophes). In addition, character literals can contain the special characters shown in Table 2.1. Examples of character literals include — *The SR Programming Language*

'a' 'Z' '4' '\''\e' '\33''\x1b'

The fourth literal is a single quote. The last three literals are all the escape (ESC) character: as a special character, i octal, and in hexadecimal. — *The SR Programming Language*

The character type has no type-specific operators, although characters can be concatenated to form strings. — *The SR Programming Language*

\n	newline (NL)	\a	alert (BEL)
----	--------------	----	-------------

- $t ab (HT) ext{e} escape (ESC)$
- b backspace (BS) v vertical tab (VT)
- r return (CR) f form feed (FF)
- $\$ single quote $\$ double quote
- \\ backslash
- **\000** bit patterns where *000* is 1-3 octal digits
- h bit pattern where *hh* is 1 or 2 hexadecimal digits
- \mathbf{C} character *c* where *c* is any other character

Table 2.1. Special characters

string types

character strings

Strings are sequences of zero or more ASCII characters enclosed in double quotes. String literals can also contain the special characters listed in Table 2.1. Examples of string literals are shown below; the first is the empty string. — *The SR Programming Language*

"" "alpha" "Z" "44" "I'm having fun" "here is a ∖" in the middle"

The string type has one binary operator, ||, for string concatenation. Each operand is either a string or a character. The result is a string that is the concatenation of the two operands. — *The SR Programming Language*

And implicit part of every string is the number of characters it holds. The declaration of a string variable must specify a maximum length, e.g., string[20]. The actual number of characters in a string variable can vary from zero up to the specified maximum. The predefined function length(s) returns the current number of characters in string S. The predefined function maxlength(s) returns the maximum number of characters in string S, e.g., the declared maximum length. For either function S can be a variable or an arbitrary string expression. The following table gives some examples of the values of these predefined functions, assuming X is a string variable declared as string[20], and its current values is "OSCOR". — *The SR Programming Language*

expr	length (<i>expr</i>)	<pre>maxlength (expr)</pre>
"abc"	3	3
x	5	20
"abc" x	8	8

The value of length and maxlength on a given expression will differ only when the expression is a single variable. — *The SR Programming Language*

bit strings

bit strings

relational operators and ordered types

relational operators and ordered types

Relational operators compare their operands and return a boolean value that relects the result of the comparison. They are defined on reals, strings, and all *ordered* types. Of the basic types, booleans, characters, and integers are ordered types; that is, it makes sense to define an ordering among values in each of these types, *and* their successive values differ by a fixed amount. Reals and strings are not considered ordered types because adjacent values differ by a variable amount. However, orderings exist between their respective values so that reals and strings cna be compared. Enumeration types, described later, are the only user-defined ordered types. *— The SR Programming Language*

The ordering among boolean values is that false precedes true. Among character values, the ordering is defined by the underlying bit representation treated as an unsigned integer. (Thus the ASCII ordering is preserved, although all eight bits of a byte can be used in a character.) The ordering of integer values by their numeric values; the same applies to real values. Strings are ordered lexicographically using their underlying character representation. The empty string is the
smallest string. For example, the following strings are in increasing order: *Language*

"" " " "12" "123" "A" "a" "alpha" "beta"

The relations operators—which again may be used only on ordered types, reals, and strings include the usual comparisons =, !=, <, <=, etc. Each operator is binary and returns a boolean result; its operands must be of the same type. The precedence of relational operators is aabove that of the boolean operators but below that of all other operators encountered thus far. Thus, expressions involving multiple relational operators can be written without requiring parenthesis (unlike in Pascal). For example, the following two boolean expressions are equivalent: — *The SR Programming Language*

a < b and c > d (a < b) and (c > d)

The relational operators = and != are also defined for pointers and for capabilities. — *The SR Programming Language*

A number of predefined functions are useful in dealing with ordered types and reals. The function min(x1,...) returns the smallest value among its arguments; the function max(x1,...) returns the largest. Both min and max take one or more arguments of the same type. For example, max('4', 'A', 'c') returns 'c'. For an ordered type T, the function low(T) returned the smallest value of type T; the function high(T) returns the largest. On a machine that stores integers in 32 bits using two's compliment form, for example, low(int) is -2147483648 and high(int) is 2147483647. The values low(real) and high(real) return, respectively, the smallest and largest representable positive real numbers. FInally, for ordered types (but not reals), the function pred(x) returns x&146;s predecessor in the defined ordering; the function succ(x) returns x&146;s successor. For example, pred(8) is 7 and succ('f') is 'g'. — *The SR Programming Language*

píctures and formats

pictures and formats

pictures and formats

pointer types

pointer types

pointer types

reference and dereference

reference and dereference

reference and dereference

access types

access types

access types

dynamic storage

dynamic storage

dynamic storage

vector types

vector types

vector types

matrix types

matrix types

matrix types

array types

array types

array types

From the <u>IBM Programmer's Reference Manual FORTRAN</u> Automatic Coding System for the IBM 704:

The FORTRAN language is intended to be capable of expressing any problem of numerical computation. In particular, it deals easily with problems containing large sets of formulae and many variables, and it permits any variable to have up to three independent subscripts.



table types

table types

table types



ranges

ranges

enumeration

types

enumeration types

enumeration types

tuples

tuples

tuples



sets

sets

structure types

structure types

structure types

union types

union types

union types

record types with discriminants

record types with discriminants

record types with discriminants

hashes

hashes

hashes

lists

lists

lists

stacks

stacks

stacks

queues

queues

queues

dictionaries

dictionaries

dictionaries

objects

objects

objects

function types

function types

function types

void types

void types

void types

user defined

types

user defined types

user definied types

files

files

files

scope and visibility

scope and visibility

scope and visibility

conversions

conversions

conversions

expressions

expressions

expressions

primary expressions

primary expressions

primary expressions

postfix

expressions

postfix expressions

postfix expressions

unary expressions

unary expressions

unary expressions

binary expressions

binary expressions

binary expressions

logical operation expressions

logical operation expressions

logical operation expressions

logic assertions

logic assertions

logic assertions
conditional expressions

conditional expressions

conditional expressions

assignment

expressions

assignment expressions

assignment expressions



sequential expressions

sequential expressions

constant

expressions

constant expressions

constant expressions

order of evaluation

order of evaluation

order of evaluation

statements

statements

statements

expression

statements

expression statements

expression statements

aríthmetic

statements

arithmetic statements

From the <u>IBM Programmer's Reference Manual FORTRAN</u> Automatic Coding System for the IBM 704:

Nature of Fortran Arithmetic

Arithmetic in the object progra will generally be performed with singleprecision 704 floating point numbers. These numbers provide 27 binary digits (about 8 decimal digits) of precisionn, and may have magnitudes between approximately 10^{-38} and 10^{38} , and zero. Fixed point arithmetic, but for integers only, is also provided.



Portions of this document are provided here. Use the link to view and read the entire document.

From the IBM Reference Manual FORTRAN II for the IBM 704 Data Processing System: As in the previous FORTRAN system, there are two kinds of arithmetic statements. The terms used in previous FORTRAN literature for the two kinds of arithmetic statements are arithmetic formula and function statement. To avoid confusion of terms, the second kind of arithmetic statement will be referred to subsequently in this manual as a function definition.



693

The formal description of the function definition is as follows:

General Form	Examples
"a=b" where a is a function name followed	FIRSTF(X) = A*X + B
(which must be distinct non-subscripted variables) separated by commas, and b is an expression which does not involve	SECONDF(X,B) = $A*X + B$
subscripted variables. The function name on the left side of the function definition	THIRDF(D) = FIRSTF(E)/D
consists of 4 to 7 alphabetic or numeric characters (not special characters), of which the last must be F and the first must be alphabetic. Also, the first must be X if	FOURTHF(F, G) = SECONDF(F, THIRDF(G))
be fixed point. Any functions appearing on the right side must be built-in, or available	FIFTHF(I, A) = 3.0*A**I
on the library tape, or already defined by preceding funciton definitions, or defined by a function subprogram	SIXTHF(J) = J + K
og a ranouon suoprorgani.	XSIXTHF(J) = J + K

labeled

statements

labeled statements

labelled expressions

blocks

blocks

blocks

compound statements

compound statements

compound statements

conditional

statements

conditional statements

conditional statements

loops

loops

loops

statements and

íteratíve

íterators

iterative statements and iterators

iterative statements and iterators

case statements

case statements

case statements

break and continue

statements

break and continue statements

break and continue statements

return

statements

return statements

return statements

goto statements

goto statements

goto statements

low-evel

statements

low-level statements

low-level statements

null statements

null statements

null statements

conditions

conditions

exceptions

exceptions

exceptions

errors

error

The language definition classifies errors into several different categories: — <u>Ada Reference Manual</u> <u>1.1.5</u>

Errors that are required to be detected prior to run time by every Ada implementation; — <u>Ada</u> <u>Reference Manual 1.1.5</u>

Errors that are required to be detected at run time by the execution of an Ada program; — <u>Ada</u> <u>Reference Manual 1.1.5</u>

Bounded errors; — Ada Reference Manual 1.1.5

The language rules define certain kinds of errors that need not be detected either prior to or during run time, but if not detected, the range of possible effects shall be bounded. The errors of this category are called *bounded errors*. The possible effects of a given bounded error are specified for each such error, but in any case one possible effect of a bounded error is the raising of the exception Program_Error. — <u>Ada Reference Manual 1.1.5</u>

Erroneous execution. — Ada Reference Manual 1.1.5

In addition to bounded errors, the language rules define certain kinds of errors as leading to erroneous execution. Like bounded errors, the implementation need not detect such errors either prior to or during run time. Unlike bounded errors, there is no language-specified bound on the possible effect of erroneous execution; the effect is in general not predictable. — <u>Ada Reference Manual</u> <u>1.1.5</u>

procedures

procedures

procedures

functions

functions

functions

recursion

recursion

recursion

methods

methods

lambdas

lambdas

input statements

input statements

input statements

From the IBM Programmer's Reference Manual FORTRAN Automatic Coding System for the IBM 704:

Provisions for Input and Output

Certain statements in the FORTRAN labguage cause the object program to be equipped with its necessary input and output programs. Those which deal with decimal information include conversion to or from binary, and permit considerable freedom of format in the external medium.



statements

output statements

output statements

From the <u>IBM Programmer's Reference Manual FORTRAN</u> Automatic Coding System for the IBM 704:

Provisions for Input and Output

Certain statements in the FORTRAN labguage cause the object program to be equipped with its necessary input and output programs. Those which deal with decimal information include conversion to or from binary, and permit considerable freedom of format in the external medium.

__ (Jr	ΓJ	2	
	AUTOMATIC	CODING	STSTEM	
	AUTOMATIC	CODING	STSTEM	
	AUTOMATIC	CODING THE IB	ststen M 704	
4	AUTOMATIC	CODING THE IB	STSTEM	

database

statements

database statements

database statements

classes

classes

classes

modules

modules

modules

packages

packages

packages

generíc uníts

generic units

generic units
reflection

reflection

reflection

operations

operations

operations

resources

resources

resources

globals

globals

globals

concurrent

execution

concurrent execution

concurrent execution

tasks

tasks

tasks

semaphores

semaphores

semaphores

asynchronous

message passing

asynchronous message passing

asynchronous message passing

728

remote procedure call

remote procedure call

remote procedure call

rendevous

rendevous

rendevous

vírtual machines

virtual machines

virtual machines

ow-eve

programming

low-level programming

low-level programming

standard library

standard library

standard library

From the <u>IBM Programmer's Reference Manual FORTRAN</u> Automatic Coding System for the IBM 704:

Inclusion of Library Routines

Pre-written routines to evaluate any single-valued functions of any number of arguments can be made available for incorporation into object programs by placing them on the master FORTRAN tape.



733

standard language additions

standard language additions

standard language additions

character processing functions

character processing functions

character processing functions

string processing functions

string processing functions

string processing functions

memory functions

memory functions

memory functions

input output functions

input output functions

input output functions

storage allocation functions

storage allocation functions

storage allocation functions

mathematical functions

mathematical functions

mathematical functions

time and date functions

time and date functions

time and date functions

control functions

control functions

control functions

míscellaneous functions

miscellaneous functions

miscellaneous functions

emergency

messag

IMPORTANT NOTE:

From Help Annie GoFundMe page:

I'm Jeff, and I'd like to help a friend, Annie, who is suffering from depression and recovering from domestic abuse. She is having a tough time keeping a roof over her head, and is scheduled to be evicted on Monday or Tuesday. So I'm trying to raise money so she can move out from the motel and get into something more stable.

She's in the state of Virginia, and me and another friend of hers (Milo) are in California and are trying to raise money to, 1 - keep a roof over her head, and 2 - get her into a better place. The Virginia Rent Relief Program was going to pay her rent, but fell through at the last minute because they won't pay for a motel.

So, the most immediate problem is paying for her motel, which is \$312 a week. Then she needs help with a first and last, etc... to move into something more stable.

I've suffered from severe depression before, and I know how horrible it is. I'd love to pay for everything, but I'm only barely getting by myself. So I'm asking for help for her.

Thank you for your time, and anything you can do to help her!

~Jeff

744