Hello, world!

ABAP

```
* ABAP: "Hello, world!" *
REPORT HELLO_WORLD.
WRITE 'Hello, world!'.
```

Ada

```
-- Ada: "Hello, world!"
with Ada.Text_IO; use Ada.Text_IO;
procedure Hello is
begin
   Put_Line ("Hello, world!");
end Hello;
```

ALGOL

```
HELLO: 'BEGIN'
   'COMMENT' CDC ALGOL 60: "Hello, world!" ;
   OUTSTRING(1, 'Hello, world!')
'END' HELLO
```

Ant

```
<!-- Ant: "Hello, world!" -->
<project name="Hello" default="hello">
  <target name="hello" description="Greeting">
    <echo message="Hello, world!"/>
  </target>
</project>
```

APL

```
X APL: Hello, world!
Y- 'Hello, world!'
```

AppleScript

```
-- AppleScript: "Hello, world!"
display dialog "Hello, world!"
```

ASP

```
<% @ Language="VBScript" %>
<% 'ASP: "Hello, world!"
   Response.Write("<html><body>Hello, world!</body></html>") %>
```

Assembler

```
; NASM Intel 8086 Assembler (DOS): "Hello, world!"
        org 100h
start:
        MOV     AH,09h
        LEA     DX,[msg]
        INT     21h
        MOV     AX,4C00h
        INT     21h
msg:    DB      'Hello, world!$'
```

awk

```
# awk: "Hello, world!"
BEGIN { print "Hello, world!" }
```

B

```
/* A TUTORIAL INTRODUCTION TO THE LANGUAGE B
B. W. Kernighan Bell Laboratories, New Jersey,
January 1973 - first documented "hello, world!" */

main( ) {
  extrn a, b, c;
  putchar (a); putchar (b); putchar (c); putchar ('!*n');
}
a 'hell' ;
b 'o, w' ;
c 'orld' ;
```

Bash

```
# Bash: "Hello, world!"
echo Hello, world!
```

BASIC

```
10 REM BASIC: "Hello, world!"
20 PRINT "Hello, world!"
```

Brainfuck

```
Brainfuck: "HelloKOMMA World!" [-]++++++++++[>++++++
+>++++++++++>+++>+++>+<<<<<<-]>++.>+.+++++++..+++.>+
+++.>++.<<++++++++.--------.+++.------.--------.>>+.
```

BS2000

```
/BEGIN-PROCEDURE LOGGING=N
/REMARK BS2000 (SDF): "Hello, world!"
/ASSIGN-SYSDTA TO-FILE=*SYSCMD
/WRITE-TEXT 'Hello, world!'
/SET-JOB-STEP
/ASSIGN-SYSDTA TO-FILE=*PRIMARY
/END-PROCEDURE
```

C

```
/* ANSI C: "Hello, world!" */
#include <stdio.h>
#include <stdlib.h>
int main(void){
    puts("Hello, world!"); return EXIT_SUCCESS;
}
```

C#

```
// C#: "Hello, world!"
using System;
class HelloWorld {
    public static int Main(String[] args) {
        Console.WriteLine("Hello, world!");
        return 0;
    }
}
```

C++

```
// ISO C++: "Hello, world!"
#include <iostream>
int main() {
    std::cout << "Hello, world!" << std::endl;
}
```

COBOL

```
      * COBOL 74: "Hello, world!"
       IDENTIFICATION DIVISION.
         PROGRAM-ID. HELLO.
       ENVIRONMENT DIVISION.
       MAIN SECTION.
         DISPLAY "Hello, world!"
         STOP RUN.
```

Common Lisp

```
;;; Common Lisp: "Hello, world!"
(defun helloworld ()
  (print "Hello, world!")
)(helloworld)
```

dBase

```
* dBase IV: "Hello, world!"
? "Hello, world!"
```

Dylan

```
// Dylan: "Hello, world!"
define method hello-world()
    format-out("Hello, world!\n");
end method hello-world;
hello-world();
```

ECMA/JavaScript

```
// ECMAScript/JavaScript: "Hello, world!"
alert('Hello, world!');
```

Eiffel

```
indexing
   description: "Eiffel: Hello, world!"
class HELLO
creation
   run
feature
   run is
     do
       print("Hello, world!%N");
     end -- run
end -- HELLO
```

Erlang

```
%% Erlang: "Hello, world!"
-module(hello).
-export([hello/0]).

hello() ->
   io:format("Hello, world!~n", []).
```

Forth

```
\ Gforth: "Hello, world!"
s" world!" s" Hello, "
type type CR
bye
```

Fortran

```
C     Fortran: Hello, world!
      PROGRAM HALLO
      WRITE (*,100)
      STOP
  100 FORMAT ('Hello, world!')
      END
```

Groovy

```
// Groovy: "Hello, world!"
println "Hello, world!"
```

Haskell

```
-- Haskell: "Hello, world!"
main :: IO ()
main = putStrLn "Hello, world!"
```

Hello

```
h
```

HPGL

```
CO HPGL: "Hello, world!";
IN0;
PA0,0;
PR1000,1000;
SP1;
PD;
DT!;
LBHello world!;
PU;
PA 0,0;
SP0;
```

HTML

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
<!-- HTML: "Hello, world!" -->
<html><head><title>Greeting</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head><body><p>Hello, World!</p></body></html>
```

INTERCAL

```
* INTERCAL: "Hello, world!"
PLEASE DO ,1 <- #13
DO ,1 SUB #1 <- #238
DO ,1 SUB #2 <- #112
DO ,1 SUB #3 <- #112
DO ,1 SUB #4 <- #0
DO ,1 SUB #5 <- #64
DO ,1 SUB #6 <- #238
DO ,1 SUB #7 <- #26
DO ,1 SUB #8 <- #248
DO ,1 SUB #9 <- #168
DO ,1 SUB #10 <- #24
DO ,1 SUB #11 <- #16
DO ,1 SUB #12 <- #158
DO ,1 SUB #13 <- #52
DO READ OUT ,1
PLEASE GIVE UP
```

Java

```
// Java: "Hello, world!"

class HelloWorld {
  public static void main(String args[]) {
    System.out.println("Hello, world!");
  }
}
```

JSP

```
<!-- Java Server Pages: "Hello, world!" -->
<%@ page language='java' %>
<%="Hello, world!" %>
```

LaTeX

```
% LaTeX: "Hello, world!"
\documentclass{article}
\begin{document}
Hello, world!
\end{document}
```

Logo

```
; Logo: "Hello, world!"
PRINT [Hello, world!]
BYE
```

Maschinencode

```
B4 09 8D 16 0D 01 CD 21  B8 00 4C CD 21 48 65 6C
6C 6F 2C 20 77 6F 72 6C  64 21 24
```

MAKE

```
# make: "Hello, World!"
all:
        @echo "Hello, World!"
```

Modula-2

```
(* Modula-2: "Hello, world!" *)
MODULE HelloWorld;
FROM InOut IMPORT WriteString,WriteLn;
BEGIN
  WriteString("Hello, world!");
  WriteLn;
END HelloWorld.
```

MS-DOS Batch

```
@ECHO OFF
REM DOS batch: "Hello, world!"
ECHO Hello, world!
```

Mumps-M

```
; Mumps-M: "Hello, world!"
 w !,"Hello, world!"
```

Oberon

```
(* Oberon: "Hello, world!" *)
MODULE Helloworld;
IMPORT Write;
BEGIN
  Write.Line("Hello, world!");
END Helloworld.
```

Pascal

```
program HelloWorld(output);
(* Pascal: Hello, world! *)
begin
  WriteLn('Hello, world!');
end.
```

Perl

```
# Perl: "Hello, world!"
print "Hello, world!\n";
```

PHP

```
<?php // PHP: "Hello, world!"
echo "<html><body>"."Hello, world!"."</body></html>"; ?>
```

PL/1

```
/* PL/1: "Hello, world!" */
Hello: procedure options(main);
       put skip list('Hello, world!');
end Hello;
```

Plankalkül

```
R1.1(V0[:sig]) => R0
R1.2(V0[:m x sig]) => R0
0 => i | m + 1 => j
[W [ i < j -> [ R1.1(V0[i: m x sig]) => R0 | i + 1 => i ] ] ]
END
R1.3() => R0
'H';'e';'l';'l';'o';',';' ';'w';'o';'r';'l';'d';'!' => Z0[: m x sig]
R1.2(Z0) => R0
END
```

PostScript

```
% PostScript: "Hello, world!"
/Courier findfont 24 scalefont setfont
100 100 moveto (Hello, world!) show showpage
```

PowerShell

```
# PowerShell: "Hello, world!"
Write-Host "Hello, world!"
```

Prolog

```
% Prolog: "Hello, world!"
?- write('Hello, world!') , nl .
```

Python

```
# Python: "Hello, world!"
print "Hello, world!"
```

REXX

```
/* REXX: "Hello, world!" */
say 'Hello, world!'
exit
```

Ruby

```
# Ruby: "Hello, world!"
puts "Hello, world!"
```

Scheme

```
;;; Scheme: "Hello, world!"
(define h(lambda()(display "\nHello, world!")(newline)()))(h)
```

sed

```
echo s | sed -ne '1s/.*/Hello, world!/p'
```

Self

```
'Hello, world!' print.
```

Simula

```
! Simula: "Hello, world!" ;
Begin
  OutText("Hello, world!"); OutImage;
End of program;
```

Smalltalk

```
'Hello, world!' out.
```

SQL

```
-- SQL: "Hello, world!"
SELECT 'Hello, world!' AS message;
```

TeX

```
% TeX: "Hello, world!"
Hello, world!
\bye
```

troff/groff

```
.\" troff/groff: "Hello, world!"
\f(CW
Hello, world!
```

Velocity

```
<!-- Velocity: "Hello, world!" --><html><body>
  #set( $foo = "Hello, world!" )
  $foo
</body></html>
```

Whitespace

XML

```
<?xml version="1.0"?><!-- XML: "Hello, world!" -->
<greet><how>Hello, </how><who>world</who><mark>!</mark></greet>
```

XUL

```
<?xml version="1.0"?><!-- XUL: "Hello, world!" -->
<window xmlns=
"http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
<label value="Hello, world!"/></window>
```

Impressum

Poster-Bestellung und mehr Informationen:
http://www.lehmanns.de/helloworld
Redaktion: Dirk Eckel, Christoph Kaeder, Matthias Stolt (Hamburg)

10623 Berlin
Hardenbergstr. 5
Tel 030/617911-0
berlin@lehmanns.de

10117 Berlin
Friedrichstraße 128
Tel 030/2827079
b-fr@lehmanns.de

12489 Berlin
Rudower Chaussee 26
Tel 030/2093-2028
adlershof@lehmanns.de

91054 Erlangen
Universitätsstraße 6
Tel 09131/829644
er@lehmanns.de

79098 Freiburg
Friedrichring 25
Tel 0761/3899010
fr@lehmanns.de

06108 Halle
Universitätsring 7
Tel 0345/212150
hal@lehmanns.de

20095 Hamburg
Kurze Mühren 6
Tel 040/336384
hh-city@lehmanns.de

50937 Köln
Universitätsstr. 20
Tel 0221/428153
k-uni@lehmanns.de

LEHMANNS FACHBUCHHANDLUNG

04109 Leipzig
Grimmaische Str. 10
Tel 0341-3397500-0
leipzig@lehmanns.de

35037 Marburg
Steinweg 35 a
Tel 06421/590120
mr@lehmanns.de

93053 Regensburg
Universitätsstraße 31
Tel 0941/90830
rgbg@lehmanns.de

89073 Ulm
Wengengasse 27
Tel 0731/63334
ulm-city@lehmanns.de

89081 Ulm
Albert-Einstein-Allee 15
Tel 0731/56600
ulm-oe@lehmanns.de

siz - Service Internationale
Zeitschriften
Tel 0800/7492665
siz@lehmanns.de

Portofreie Lieferung
innerhalb Deutschland
Kostenlose Service Hotline
0800-2662665 (COMBOOK)

Email Newsletter
24 Themen Ihrer Wahl
Subscribe/Unsubscribe unter:
www.lehmanns.de/email-news