

Schriftgröße S

ActionScript

```
1 package de.hwh.bsp.hallowelt
2 {
3     import flash.display.Sprite;
4     import flash.events.Event;
5
6     public class Main extends Sprite
7     {
8         public function Main()
9         {
10            if (stage)
11                init();
12            else
13                addEventListener(Event.ADDED_TO_STAGE,
14            }
15
16            private function init(e:Event = null):void
17            {
18                removeEventListener(Event.ADDED_TO_STAGE,
19
20                trace("Hallo Welt!");
21            }
22        }
23 }
```

Hallo Welt!

AppleScript

```
1 say "Hallo Welt"
2 display dialog "Hallo Welt"
```

Hallo Welt!

Arduino

```
1 // Pin 13 has an LED connected on most Arduino boards.
2 // give it a name:
3 int led = 13;
4
5 // the setup routine runs once when you press reset:
6 void setup() {
7   // initialize the digital pin as an output.
8   pinMode(led, OUTPUT);
9 }
10
11 // the loop routine runs over and over again forever:
12 void loop() {
13   digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
14   delay(1000); // wait for a second
15   digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
16   delay(1000); // wait for a second
17 }
```

Arduinos kommunizieren bevorzugt mit anderen elektronischen Komponenten. Ein einfaches Minimal-Beispiel ist also beispielsweise eine einzelne LED zum Blinken zu bringen. Quelle: <https://www.grundwissen.de/elektronik/arduino/erste-beispiele.html>

Bash

```
1 foo="Hallo, Welt"
2 echo "$foo"
```

Bash-Skripte sind geeignet um Programme, gemäß der eigenen Bedürfnisse, miteinander zu kombinieren und sie spielen eine zentrale Rolle im Alltag der Systemadministration.

BASIC

```
1 cls
2 print "Hello World in BASIC!"
3 sleep
4 end
```

Ein Hello World!, auch bekannt als Hallo Welt!, wird traditionell verwendet, um Anfänger in eine Programmiersprache, zum Beispiel BASIC, einzuführen.

Clojure

```
1 bash-3.2$ lein repl
2 nREPL server started on port 59553 on host 127.0.0.1
3 REPL-y 0.2.1
4 Clojure 1.5.1
5   Docs: (doc function-name-here)
6         (find-doc "part-of-name-here")
7   Source: (source function-name-here)
8   Javadoc: (javadoc java-object-or-class-here)
9   Exit: Control+D or (exit) or (quit)
10  Results: Stored in vars *1, *2, *3, an exception in *e
11
12 user=> (prn "Hello World")
13 "Hello World"
14 nil
15 user=> (println "Hello World")
16 Hello World
17 nil
18 user=> (pr-str "Hello World")
19 "\"Hello World\""
```

This is always the first tiny baby step when learning a programming language. Let's try "Hello, World!" in Clojure.

CMake

```
1 > cmake --build .
2 Scanning dependencies of target prog
3 [ 50%] Building CXX object CMakeFiles/prog.dir/main.cpp.obj
4 [100%] Linking CXX executable prog.exe
5 [100%] Built target prog
6
7 > ./prog.exe
8 Hello CMake!
```

CMake is one of the most popular build systems for C++ out there. One of the main reasons probably is that it is cross-platform: It does not build the project itself but operates a platform-specific system.

Schriftgröße M

CoffeeScript

```
1 coffee> console.log "Hello World!"
```

CoffeeScript ist eine relativ junge Sprache, die nach JavaScript kompiliert wird. CoffeeScript hat eine goldene Regel: Es ist einfach JavaScript

C++

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     cout << "Hallo, Welt!" << std::endl;
6     return 0;
7 }
```

C++ (zeh plus plus ausgesprochen) ist eine kompilierte Allzweck Programmiersprache. Ihre Komplexität liegt zwischen Einsteiger und Fortgeschritten, da sie sowohl High-Level als auch Low-Level Sprachelemente vereint. Sie bietet imperative, objekt-orientiert

C-Sharp (C#)

```
1 using System;
2 namespace HelloWorldApp {
3     class Geeks {
4         static void Main(string[] args)
5             Console.WriteLine("Hello World!");
6             Console.ReadKey();
7     }
8 }
9 }
```

In C#, a basic program consists of the following: A Namespace Declaration, Class Declaration & Definition, Class Members (like variables, methods etc.), Main Method, Statements or Expressions.

CSS

```
1 <style type="text/css">
2 h1 {
3   color: DeepSkyBlue;
4 }
5 </style>
6
7 <h1>Hello, world!</h1>
```

Bring the text "Hello, world!" to the user's screen could easily be accomplished in pure HTML, without the use of any CSS, so we'll spice it up just a bit with a different color:

Dart

```
1 void main() {
2   print('Hello, World!');
3 }
```

This collection is not exhaustive—it's just a brief introduction to the language for people who like to learn by example.

Delphi

```
1 Label1.Caption := 'Hello world';
```

Auf die neue Form setzt man nun einen Button und ein Label (beides Registerkarte Standard). Wenn man nun doppelt auf den Button klickt, öffnet sich das Code-Fenster. Hier geben Sie folgendes ein:

Elixir

```
1 # module_name.ex
2 defmodule ModuleName do
3   def hello do
4     IO.puts "Hello World"
5   end
6 end
```

This is a quick introduction to the Elixir syntax for Erlang developers and vice-versa.

Schriftgröße L

Elm

```
1 module HelloWorld exposing (..)
2
3 import Html exposing (text)
4
5 main =
6   text "Hello, World!"
```

And, that's it! We can write Hello World in Elm in just a few lines of code, but what's really going on in this code?

Erlang

```
1 % module_name.erl
2 -module(module_name). % you may use some other name
3 -compile(export_all).
4
5 hello() ->
6   io:format("~s~n", ["Hello world!"]).
```

when you want to define a named function, Erlang expects it to be inside of a module, and modules have to be compiled.

Excel

```
1
2 Sub helloWorld()
3
4   MsgBox ("Hello World!")
5
6 End Sub
7
8 Sub helloWorldInZelle()
9
10  Sheets("Tabelle1").Select
11
12  Cells(1, 1).Value = "Hello World!"
13
14 End Sub
```

Mit dem Tabellenkalkulationsprogramm Excel könnt Ihr nicht nur Diagramme und Statistiken erstellen sondern habt auch die Möglichkeit im Hintergrund komplexe Programme mit VBA (Visual Basic for Applications) zu programmieren.

F#

```
1 open System
2
3 [<EntryPoint>]
4 let main argv =
5     printfn "Hello World"
6     Console.ReadLine() |> ignore
7     0
```

programmers well know, it is customary to start with a “Hello World” example. So we will be doing just that. So without further ado, what does it take to create a stand alone “Hello World” app in F#.

Go

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     fmt.Println("hello world")
7 }
```

Our first program will print the classic “hello world” message. Here’s the full source code.

Groovy

```
1 println 'Hello Groovy-World!'
```

Zwei Merkmale sind hier auf jeden Fall zu nennen und insbesondere die Kombination aus beiden macht Groovy im Moment so einzigartig:
ausdrucksstarke, prägnante Syntax und dynamische Eigenschaften
perfekte Integration mit Java.

Haskell

```
1 Prelude> putStrLn "Hello World"
2 Hello World
```

Haskell is a general purpose, purely functional programming language. This page will help you get started as quickly as possible.

Schriftgröße XL

Haxe

```
1 class Main {
2     static public function main():Void {
3         trace("Hello World");
4     }
5 }
```

The following program prints "Hello World" after being

Java

```
1 public class HelloWorld
2 {
3
4     public static void main (String[] args
5     {
6         // Ausgabe Hello World!
7         System.out.println("Hello World!");
8     }
9 }
```

In Java programmiert man aufgrund der zugrunde liegenden

JavaScript

```
1 <script>
2     alert( 'Hello, world!' );
3 </script>
```

JavaScript programs can be inserted into any part of an HTML document with the help of the <script> tag.

JSON

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Test Javascript</title>
5     <script type="text/javascript">
6       let hello_world = {"Hello":"World"};
7       alert(hello_world.Hello);
8     </script>
9   </head>
10  <body>
11    <h2>JSON Hello World</h2>
12    <p>This is a test program to alert Hello
13  </body>
```

Julia

```
1 println("hello world")
2 #> hello world
```

Set of unofficial examples of Julia the high-level, high-performance dynamic programming language for technical computing.

Kotlin

```
1 fun main(args: Array<String>) {
2     println("Hello World!")
3 }
```

Let's get straight to the point - type this into a file with the extension .kt:

LaTeX

```

1 Man kann auch ganz andere Geräte (Ha, der ers
2 Esszett!) referenzieren, zum Beispiel die fun
3 \ref{Gleichung}.
4 \begin{equation}\label{Gleichung}
5   \int^{\infty}_{-\infty} e^{-x^2}dx = \sqrt{\pi}
6 \end{equation}

```

Let's get straight to the point - type this into a file with the

Lisp

```

1 (defun f(x) (/
2           (+ (* x x) 3)
3           (- a x)))

```

Set of unofficial examples of Julia the high-level, high-performance dynamic programming language for technical computing.

Lua

```

1 #!/usr/local/bin/lua
2 -- Lua Code für die Ausführung

```

Now that we have a basic understanding of JSON, let us work on our Hello World program.

Markdown

```

1 Dieses Wort ist fett.

```

JavaScript programs can be inserted into any part of an HTML document with the help of the `<script>` tag.

Weitere Beispiele für Sprachen

MATLAB

```
1 bar(b)
2 xlabel('Sample #')
3 ylabel('Pounds')
```

Matlab Beispielcode

Objective-C

```
1 output = [object methodWithOutput];
2 output = [object methodWithInputAndOutput:input];
```

Objective-c Beispielcode

OCaml

```
1 # let square x = x * x;;
2 val square : int -> int = <fun>
3 # square 3;;
4 - : int = 9
5 # let rec fact x =
6     if x <= 1 then 1 else x * fact (x - 1);;
7 val fact : int -> int = <fun>
8 # fact 5;;
9 - : int = 120
10 # square 120;;
11 - : int = 14400
```

OCaml Beispielcode

PHP

```
1 cd ../php-x.x.x
2 ./configure --enable-fpm --with-mysql
3 make
4 sudo make install
```

PHP Beispielcode

Text

```
1 {{Beispiel|zeige=Beispiel:Beispiel.txt|
2 {{BeispielCode|...}}
3 }}
```

Text Beispielcode

Python

```
1 laenge_zoll = input("Bitte Länge in Zoll eingeben:")
2 laenge_cm = laenge_zoll * 2.54
3 print "Ergebnis:", laenge_cm , "Zentimeter"
```

Python Beispielcode

Ruby

```
1 some_list.each do |item|
2   # Wir sind im Innern des Blocks
3   # und arbeiten mit item.
4 end
```

Ruby Beispielcode

Rust

```
1 cat << EOF > ./rust-tutorial.rs
2 fn main() {
3     println!("Hello, World!");
4 }
5 EOF
```

Rust Beispielcode

Scala

```
1 > scala -classpath . HalloWelt
2 Hallo, Welt!
```

Scala Beispielcode

Scheme

```
1 (define a-number 42)
2
3 (define square
4   (lambda (x)
5     (* x x)))
```

Scheme Beispielcode

Shell

```
1 char shellcode[] =
2 "\xeb\x2a\x5e\x89\x76\x08\xc6\x46\x07\x00\xc7\x46\x0c\x00\x00\x00"
3 "\x00\xb8\x0b\x00\x00\x00\x89\xf3\x8d\x4e\x08\x8d\x56\x0c\xcd\x80"
4 "\xb8\x01\x00\x00\x00\xbb\x00\x00\x00\xcd\x80\xe8\xd1\xff\xff"
5 "\xff\x2f\x62\x69\x6e\x2f\x73\x68\x00\x89xec\x5d\xc3";
```

Shell Beispielcode

SQL

```
1 CREATE TABLE <Tabellenname> (
2     <Spaltenname> <Datentyp> [ NOT NULL ] ,
3     ..... ,
4     <Spaltenname> <Datentyp>
5 ) ;
```

SQL Beispielcode

Swift

```
1 println("Hello, world!")
```

Swift Beispielcode

TypeScript

```
1 const user = {
2     name: "Hayes",
3     id: 0,
4 };
```

TypeScript Beispielcode

Visual Basic .NET

```
1 ' Mehrfach genutzte Prozedur
2 Private Sub AlleSehen()
3     ' Ausnahmebehandlung bei Zugriffsproblem
4     Try
5         ' Verbindung öffnen
6         con.Open()
7         ' SQL-Befehl zum Abruf aller Daten erstellen
8         cmd.CommandText = "SELECT * FROM personen"
9         ' Aufruf einer Prozedur zur Ausgabe
10        Ausgabe()
11    Catch ex As Exception
12        MessageBox.Show(ex.Message)
13    End Try
14    ' Verbindung schließen
15    con.Close()
16 End Sub
```

Visual Basic.NET Beispielcode

Visual Basic Script

```
1 Dim ObjShell
2 Dim ShellObject
3 Set ShellObject = CreateObject("WScript.Shell")
4 Set ObjShell = ShellObject.RegRead("HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows
5 If ObjShell = "" then
6     ShellObject.Popup "Wert existiert nicht und wird hinzugefügt", "4", ""
7     ShellObject.RegWrite "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVe
8 else
9     ShellObject.Popup "Wert existiert", "3", ""
10 end if
```

Visual Basic Script Beispielcode

Visual Basic Script HTML

```
1 <SCRIPT LANGUAGE="VBScript">
2 <!--
3 Function KannLiefern(Dt)
4 KannLiefern = (CDate(Dt) - Now()) > 2
5 End Function
6 -->
7 </SCRIPT>
```

Visual Basic Script HTML Beispielcode

XML

```
1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <bookstore>
3   <book category="cooking">
4     <title lang="en">Everyday Italian</title>
5     <author>Giada De Laurentiis</author>
6     <year>2005</year>
7     <price>30.00</price>
8   </book>
9   <book category="children">
10    <title lang="en">Harry Potter</title>
11    <author>J K. Rowling</author>
12    <year>2005</year>
13    <price>29.99</price>
14  </book>
15  <book category="web">
16    <title lang="en">Learning XML</title>
17    <author>Erik T. Ray</author>
18    <year>2003</year>
19    <price>39.95</price>
20  </book>
21 </bookstore>
```

XML Beispielcode

YAML

```
1 database_driver:  pdo_pgsql
2 database_host:   localhost
3 database_port:   5432
4 database_name:   mapbender
5 database_path:   ~
6 database_user:   postgres
7 database_password: geheim
```

YAML Beispielcode