FORTRAN:

```
C     Hello World in FORTRAN

      PROGRAM HELLO
      WRITE (*,100)
      STOP
  100 FORMAT (' Hello world! ' /)
      END
```

COBOL:

```
******************************
IDENTIFICATION DIVISION.
PROGRAM-ID. HELLO.
ENVIRONMENT DIVISION.
DATA DIVISION.
PROCEDURE DIVISION.
MAIN SECTION.
DISPLAY "Hello world!"
STOP RUN.
******************************
```

Common LISP:

```
(defun helloworld ()
  (print "Hello world!"))
```

Scheme:

```
(define hello
  (lambda ()
    (display "Hello world!")
    (newline)))
```

BASIC:

```
10 PRINT "Hello world!"
20 END
```

C:

```
#include <stdio.h>

int main(void) {
  printf("Hello world!\n");
  return 0;
}
```

C++:

```
#include <iostream.h>

main()
{
    cout << "Hello world!" << endl;
    return 0;
}
```

C#:

```
using System;

class HelloWorld
{
    public static int Main(String[] args)
    {
        Console.WriteLine("Hello world!");
        return 0;
    }
}
```

Java:

```
public class Hello {

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

Linux Assembly Language:

```
        SECTION .data
        msg     db      "Hello, world!",0xa ;
        len     equ     $ - msg
        SECTION .text
        global main
main:
        mov     eax,4
        mov     ebx,1
        mov     ecx,msg
        mov     edx,len
        int     0x80
        mov     eax,1
        mov     ebx,0
        int     0x80
```

Windows Assembly Language:

```
TITLE Hello world in win32. Tasm

VERSION T310
Model use32 Flat,StdCall

start_code segment byte public 'code' use32
begin:
 Call MessageBox, 0, offset sHallo, offset caption, 0
 Call ExitProcess, 0
start_code Ends

start_data segment byte public 'data' use32

sHallo  db 'Hello world',0
caption db "Hi",0

start_data Ends
End begin
```

Befunge:

```
>              v
v  ,,,,,"Hello"<
>48*,          v
v,,,,,,"World!"<
>25*,@
```

Python:

```
def hello():
   print "Hello world!"
```