

# Hello World

## Syntax in C

The rules that dictate the correct format of code for a specific programming language are known as syntax.

Examples of syntax in C are:

- All statements must end with a semicolon, ;
- Keywords and other code elements are case-sensitive

When compiling C code, an error will occur when the syntax of the code is incorrect.

```
// Statements must end in a semicolon (;)
// correct
printf("Hello World!");
```

```
// error
printf("Hello World!");
```

```
// Code elements are case sensitive
// correct
printf("Hello World!");
```

```
// error
PRINTF("Hello World!");
```

## Escape Sequences

In C, an escape sequence is a non-visual character used within a string.

`\n` is an escape sequence that adds a newline to a string. `\t` is an escape sequence that adds a tab of spaces to a string.

```
// \n acts as a newline in a string
printf("Hello\nWorld!"); // Outputs:
Hello

World!
```

```
// \t acts as a tab in a string
printf("Hello\tWorld!"); // Outputs:
Hello    World!
```

## Comments in C

In C, comments are text within code that will be ignored by the compiler. They are used to document code.

Line comments begin with a double forward slash, `//`. All text after `//` will be part of the comment until a new line is reached.

Block comments begin with a forward slash and asterisk, `/*` and end with an asterisk and forward slash, `*/`. Block comments can span multiple lines as new lines are part of the comment.

```
// Comments
```

```
/* This review content is
about comments and how they
can be used to document code */
```

```
// This is a line comment
```

```
/* This is a
```

## Compiling C Code with gcc

`gcc` is an application used to compile C programs into an executable that can run on the target computer.

`gcc` stands for GNU Compiler Collection.

`gcc` compiles C code using the code file as an unflagged command-line argument. The output executable file will be called `a.out`. The `-o` flag followed by some text can be used to designate the name of the output executable file.

```
gcc script.c
```

```
gcc script.c -o myProgram
```

 [Print](#)    [Share ▼](#)