In this topic, you will learn how to develop your first Python programs. Even though these programs are quite simple, they are still syntactically correct and show that programming in Python is a treat.

The Hello World program

Our first example will be **Hello, World!** It is traditionally used to introduce beginners to a new programming language.

```
print("Hello, World!")
```

As you can see, this program consists of a single line. It prints the string passed in parentheses but without quotes. You may run this code online. To that end, copy the code to this website and click on the triangle. Alternatively, follow these installation tips. You should get this result:

```
Hello, World!
```

Although this code is very simple, we will thoroughly review it.

Short explanation

print is the name of a function. A **function** is a block of code that does useful work for you, e.g., printing text. In a sense, a function is a subprogram that can be reused within your programs. When the name of a function is followed by parentheses, it means the function was **called** to get the result.

Let's go further:

```
"Hello, World!"
```

is a Python string. All strings are surrounded by *single* or *double* quotes, so

```
'Hello, World!'
```

is also a valid string. You may replace this string with another one, and the program will print the new string. For example:

```
print('Python 3.x')
```

As you might guess, this program will print the following:

```
Python 3.x
```

Printing quotes

Imagine that the string you want to print already contains quotation marks. If you would like to include them into a string, enclose this string in quotes of **another** type, e.g.:

```
print("Yes, I'm ready to learn Python.")
```

The part of the string with

```
I'm
```

is printed **correctly** because you used double quotes

```
"..."
```

to enclose the whole string:

```
Yes, I'm ready to learn Python.
```

Writing the following is **wrong**:

```
print('Yes, I'm ready to learn Python.')
```

Your program won't know where the string starts and ends.

You can run all examples using the website provided earlier. This will help you familiarize yourself with Python.

Possible errors

Even the simplest lines of code may contain errors. The most common of them are:

- **Putting extra indentation**

```
  print("Hello, World!")
```

This statement does not work because of extra spaces before

`print`

.

- **Calling a function by the wrong name**

```
pint("Hello, World!")
```

This line contains

`pint`

instead of

`print`

. Make sure to refer to every function by its proper name.

- **Writing names in the wrong case**

```
PRINT("All caps")
```

`Print`

,

`print`

and

`PRINT`

are not the same. Names are case-sensitive in Python.

- **Missing one or both quotes for a string**

```
print("Python)
```

This statement does not work because of missing closing quotes.

- **Missing one or more parentheses**

```
print("I have no end"
```

Be careful with parentheses, especially when calling a function.

With the above information, you shouldn't have any serious trouble with such programs.

Summary

In this topic, we've written our first program in Python. We broke it down, printed some strings, and looked at the most common errors you may encounter initially.

Command that makes computer performs a specific task in the command-line interface.

Function that takes arguments in parenteses, performs a specific task for you, and return results.

Statement that performs actions, but doesn't return anything, and doesn't require parenteses.

Expression that consist only of values (such as 2 or 'hello') and operators (such as + or ==).