

Theory: Overview of the basic program

🕒 6 minutes 8 / 8 problems solved

Unskip this topic

Start practicing

46352 users solved this topic. Latest completion was 16 minutes ago.

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

§1. The Hello World program

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

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

As you can see, it consists of a single line and just prints a string passed in the parentheses, but without quotes. You may run this code [online](#) (just copy it and click on the triangle) or follow these [installation tips](#). You should get this result:

```
1 | Hello, World!
```

Although this code is very simple, we will go through it in some more detail.

§2. Short explanation

Here, **print** is the name of a function. A **function** is a block of code that does some useful work for you, e.g. prints a text. In some 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 that it was **called** to get the result.

Let's go further, **"Hello, World!"** is a Python string. All strings are surrounded by either *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:

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

As you might guess, this program will print:

```
1 | Python 3.x
```

§3. Printing quotes

Now imagine that the string you want to print already contains some type of quotation mark. If you would like to include quotes into a string, then enclose this string in quotes of **another** type, e.g.:

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

Part of the string with **I'm** is printed **correctly**, because you used double quotes **"..."** to enclose the whole string:

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

If you write in the following **wrong** way:

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

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

You can try to run all the examples using the [link](#) provided earlier. This will help you familiarize yourself with Python.

Current topic:

✓ [Overview of the basic program](#) Stage 1 18★ ...

Topic depends on:

✓ [Introduction to Python](#) Stage 1 18★ ...

Topic is required for:

✓ [Multi-line programs](#) Stage 1 18★ ...

✓ [PEP 8](#) Stage 1 17★ ...

Table of contents:

[1 Overview of the basic program](#)

[§1. The Hello World program](#)

[§2. Short explanation](#)

[§3. Printing quotes](#)

[§4. Possible errors](#)

[Feedback & Comments](#)

§4. Possible errors

Even this simple line of code may contain errors, most common of them are:

- putting extra indentation

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

This does not work because of extra spaces before `print`.

- calling the function by the wrong name

```
1 | 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

```
1 | PRINT("All caps")
```

Again, `Print`, `print` and `PRINT` are not the same. Names are case-sensitive in Python.

- missing one or both quotes for a string

```
1 | print("Python)
```

This does not work because of missing closing quotes.


- missing one or more parentheses

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

Be careful with parentheses, especially when calling a function.

Now you shouldn't have any serious trouble with such programs.

 Report a typo

 Thanks for your feedback!

Start practicing

[Comments \(11\)](#)

[Hints \(0\)](#)

[Useful links \(1\)](#)

[Show discussion](#)