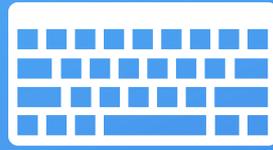


CODEFATHER



THE PYTHON

STARTER

CHECKLIST



ABOUT CODEFATHER

Codefather is committed to transform the knowledge of aspiring developers worldwide. Our mission is to guide professionals in their growth and support them in fulfilling their potential.

WE WANT TO HELP YOU

Get Started With Programming Fast

Become Confident About Your Programming Skills

Create or Transform Your IT Career

Be the Go-to Person in Your Working Environment

CODEFATHER

©Copyright 2020

CodeFather. All Rights Reserved.

hello@codefather.tech

Are you thinking about starting with programming but you don't know what the first step should be?

Are you a little bit confused about which programming language to choose?

That's okay, this is the right place where to start. Use this checklist as a step-by-step process that will help you get started with programming using one of the most popular programming languages worldwide: **Python**.

Python is one of the best languages to learn when you are starting with programming. Its learning curve is low compared to other languages like Java or C++ and with Python you can start writing your own programs fast.

This checklist is a 5-step process to start programming in Python and to start building the foundations of your programming knowledge.

Let's get started!

Claudio Sabato



STEP 1: COMPUTERS CAN SPEAK

PYTHON INTERPRETER AND PRINT STATEMENT

Programming is a way for you to talk to computers using a language that computers understand. To do that you write code that gets translated into **machine language** (the only language that computers speak and that ultimately is made of **sequences of 0s and 1s**).

When you write a Python program it gets translated into machine language by the **Python interpreter**.

The good news is that in the same way you can talk to computers, they can also talk to you. How?

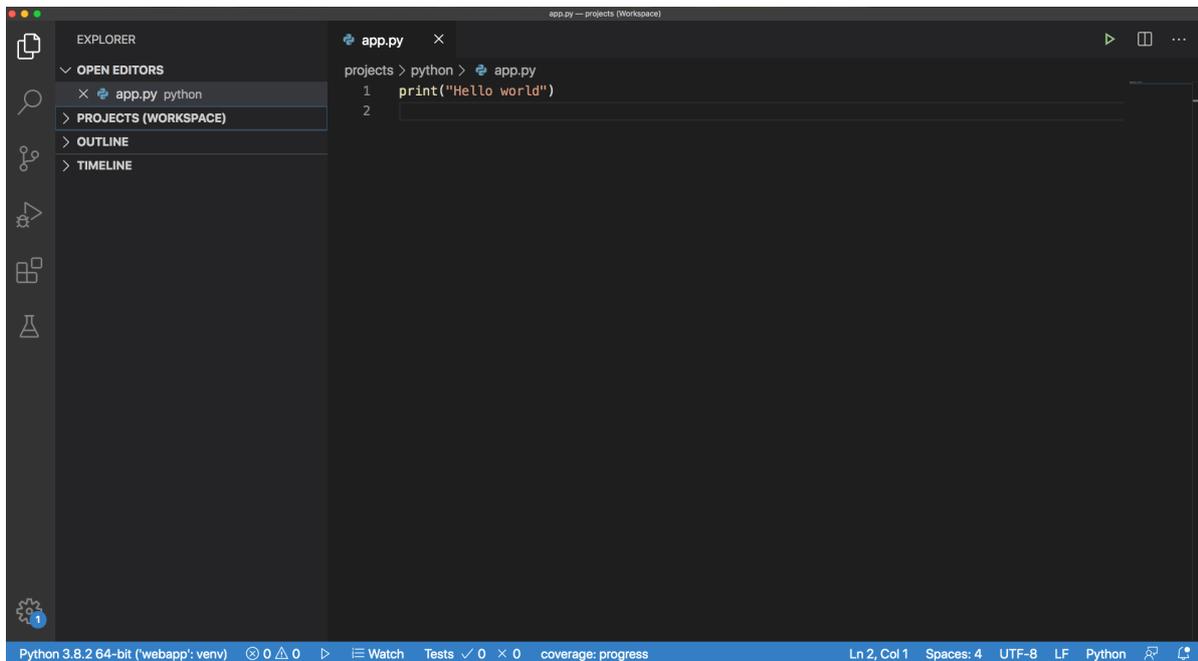
In Python this is done using the **print function**. The term function refers to code written to perform a specific operation, in this case printing text on the screen.

Let's see what this means in practice. To write code in Python (and in any other programming language) you should install on your computer an IDE (**Integrated Development Environment**). This is basically a software that allows you to write and test the code you write and it also simplifies the experience of writing code (e.g. using autocompletion while you write your code).

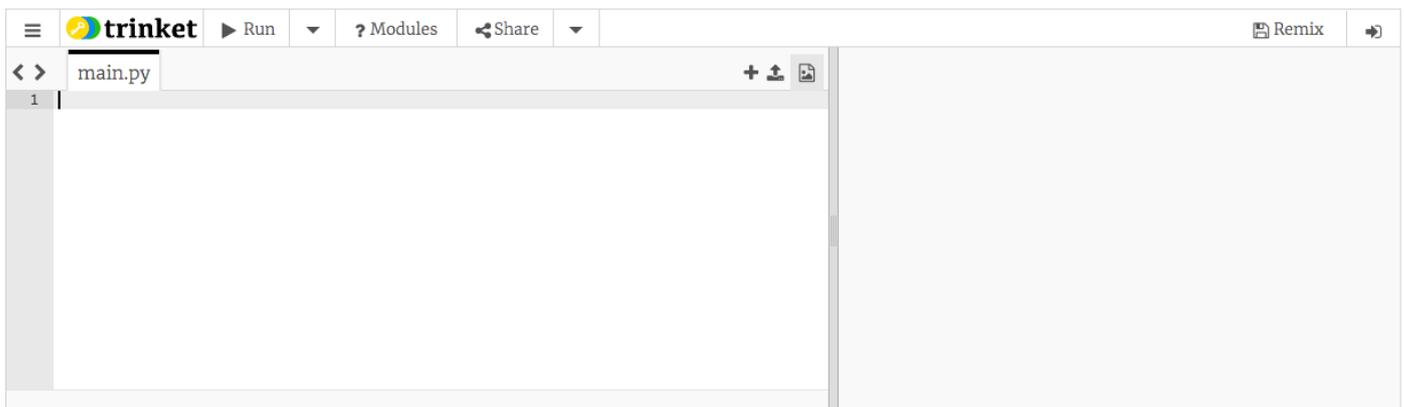


THE PYTHON STARTER CHECKLIST

Two famous IDEs that can be used for Python development are **Visual Studio Code** and **PyCharm**. Below you can see a screenshot from Visual Studio Code:



To make things simpler for this checklist, I have created an online Python interpreter that you can use directly on the Codefather website without installing anything on your computer. [You can find it here.](#)



On the **left side** of the online Python interpreter you can write your code and on the **right side** you will see the result after clicking on the **Run** button.

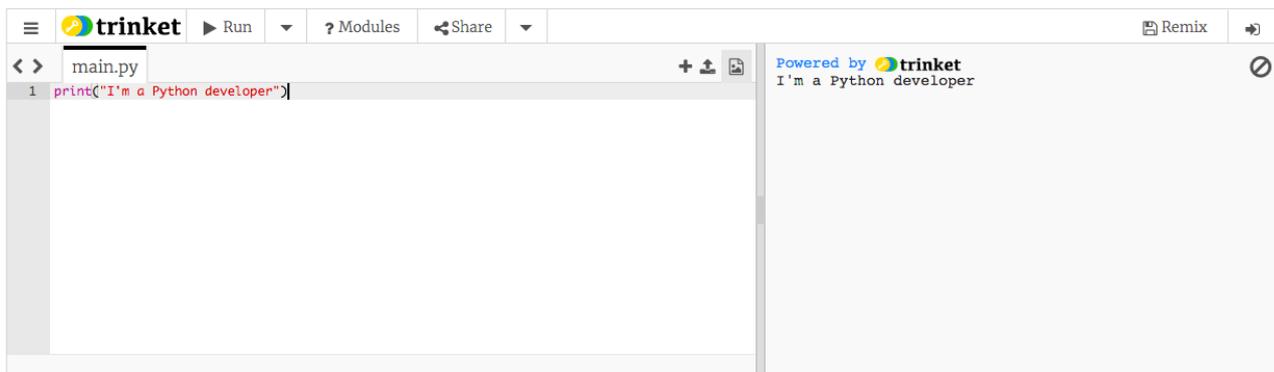
Now, let's go back to the print function....

We will use it to print the message **"I'm a Python developer"**.

On the left side of the online interpreter type:

```
print("I'm a Python developer")
```

and then click **Run**. You will see the message on the right side.



This is the way for you to test any Python code in this checklist and to get familiar with common programming constructs.

Note: The print function uses parentheses in **Python 3** (the version of Python we will work with) and doesn't require parentheses in **Python 2**.



STEP 2: A PLACE FOR YOUR DATA

VARIABLES

Variables are used to store temporary data required for the execution of a program. For example, variables can be used in Python to store text, numbers and other types of data.

To store text in a variable you use a variable of type **string**, to store numbers you use different types of variables depending on the type of number. Two examples are the **integer** (to store signed integers) and the **float** (to store floating point numbers).

A value is assigned to a variable using the equal sign (=).

Here is how to assign the value to the **variable message** of type string. A string is enclosed in single or double quotes:

```
message = "I'm a Python developer"
```

Let's assign the value 5 to the **variable count** of type integer:

```
count = 5
```

If we go back to the print statement, we can now update our program to print the value of the variable message.



Our program becomes:

```
message = "I'm a Python developer"  
print(message)
```

Run the program in the online interpreter, you will see the exact same output as before.



STEP 3: EXPLAIN WHAT YOU WANT TO DO

WRITING COMMENTS

An important programming practice is adding comments to your code. Comments explain what a program does to help you:

- Go back to programs you have written (if you write a program, and you go back to it one year later, you may doubt you wrote it because the way we code changes over time).
- Understand programs written by other developers without trying to figure out what was in their head when they wrote the program.

You can identify comments by looking at lines that start with the character # or that contain the character #.

The following are both valid comments:

```
# Print a message  
message = "I'm a Python developer"  
print(message)
```

```
message = "I'm a Python developer"  
print(message) # Print a message
```



STEP 4: PYTHON IS FLEXIBLE

DYNAMIC DATA TYPES

Explicit variable declaration is not required in Python (differently from programming languages like Java).

Let's say we want to create the variable `date` that contains a string with the current date:

Python:

```
date = "01/01/2020"
```

Java:

```
String date = "01/01/2020";
```

Can you see the difference?

In Python we are not telling the interpreter that `date` is a string, the interpreter automatically understands it from the fact that we have assigned `"01/01/2020"` to it.

Update the program to print the message: "Today's date is 01/01/2020" using the value of the variable `date`.



Here is the full program:

```
# Print a message
message = "I'm a Python developer"
print(message)

date = "01/01/2020"
print("Today's date is " + date)
```

This time in the parentheses of the print function we have also specified the variable date and we have used the **plus (+) to concatenate** the string "Today's date is" to the value of the variable date.

Write this code in the online interpreter and make sure it prints the output you expect.

Try not to copy / paste the code. Instead write it from scratch, that's the best way to learn fast.

Does the code work?



STEP 5: PYTHON MIGHT NOT LIKE YOUR CODE

UNDERSTANDING ERRORS

Let's modify our program and remove the closing parentheses at the end of the last line:

```
# Print a message
message = "I'm a Python developer"
print(message)

date = "01/01/2020"
print("Today's date is " + date
```

When we run the program we see a `SyntaxError` that tells us that there is a problem in our program.

Where does this error come from?

The interpreter executes our Python code and returns syntax errors if the way our program is written doesn't follow standard Python syntax rules.

In the error the online interpreter also says that the error is related to the file `main.py`.

The default extension for files that contain Python programs is `.py`.



BONUS STEP: FIND THE ERROR

GETTING FAMILIAR WITH PYTHON

Below you can see a Python program that contains one error.

Execute the program to see the error returned by the interpreter, try to fix it and execute the program again.

Note: in this program we have **three integer variables**: a, b and c.

```
# Calculate the sum of two numbers
```

```
a = 4
```

```
b = 5
```

```
c = a + b
```

```
print("The sum is " + c)
```

Hint: The Python **str(x)** function can be used to convert a **number x** in its string representation.

Have you found the error? And the fix?

Send me an email to hello@codefather.tech and let me know what do you think it's causing the error and what the fix is.



WHAT'S NEXT?

You have now broken the ice with programming and Python.

Well done!

How did you find it?

To start learning more about Python have a look at the [articles about Python programming](#) I publish regularly on the **Codefather** blog.

And if you prefer learning from videos subscribe to the [Codefather Youtube channel](#).

If you have any questions or if you need any support feel free to email me at **hello@codefather.tech**.

To your knowledge!

Claudio Sabato

