Python Basics

The Print Statement

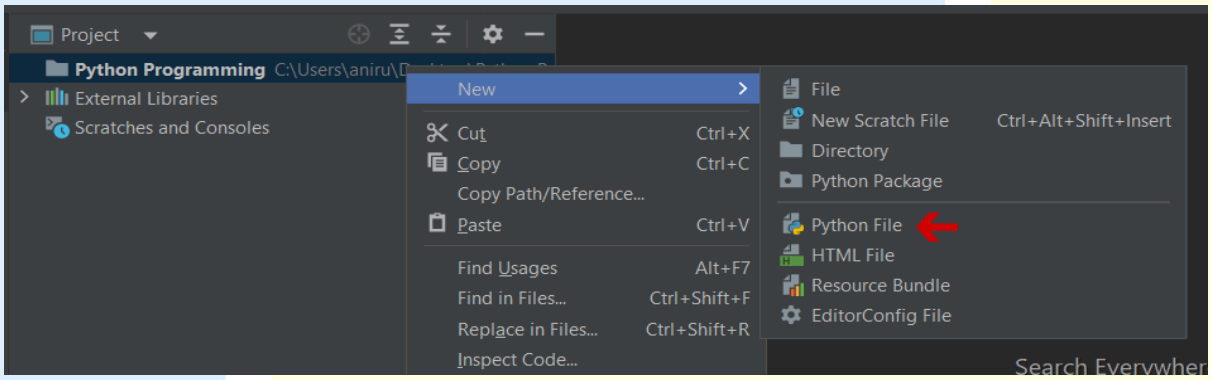
print() is a function in Python that allows us to display whatever is written inside it. In case an operation is supplied to print, the value of the expression after the evaluation is printed in the terminal.

Every running program has a text output area called "standard out", or sometimes just "stdout". The Python print() function takes in python data such as ints and strings, and prints those values to standard out.

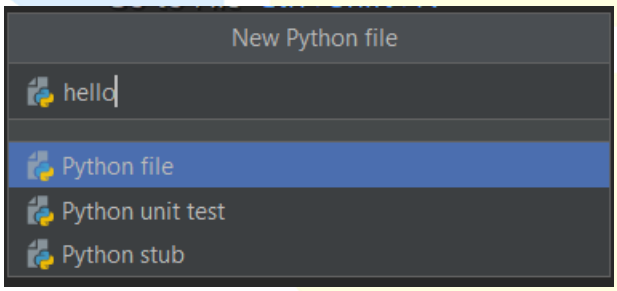
Let us start by writing and running our first code.

Delete the **main.py** file by clicking on the file and pressing the **DELETE** key.

Create a new file by right clicking on folder name -> New -> Python File.



Type in whatever name you want to give your file as per your preference and press Enter.

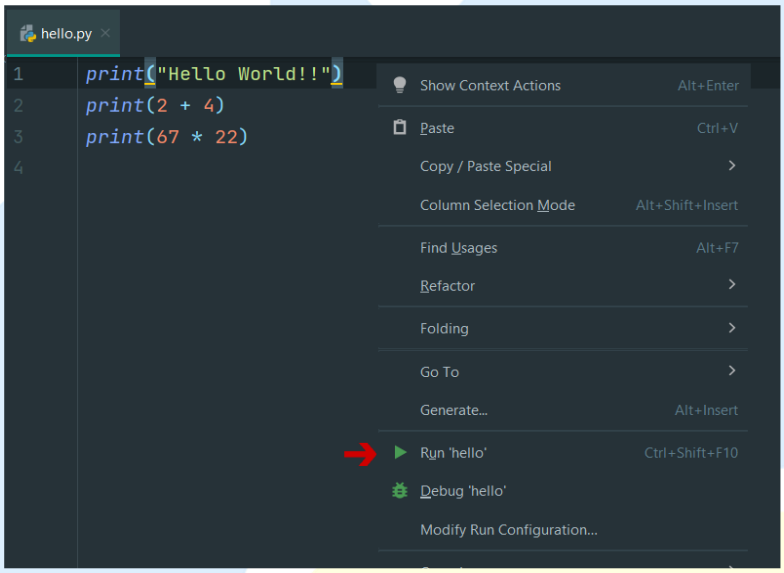


Write the following code and try to run it.

```
print("Hello World!!")
print(2 + 4)
print(67 * 22)
```

When you check the output, you can see we can print anything we want in terminal using the **print** statement. Also, we can do calculations using Python.

To run the program, right click on file and click **run.** In my case, file name is **hello.py**, it would be same as the name you have given while creating the file.



You will get the following output.

```
Hello World!!
6
1474
```

Print using END Parameter

It helps us to specify what we need to print at the end of print statement. By default, the print statement applies the **Next Line** to every statement. Check the code below to have better understanding.

```
print("Hello World")
print("This is another print statement")
```

Output:

```
Hello World
This is another print statement
```

Now let us use the **END** parameter and check the output.

Example 1:

```
print("Hello World", end="")
print("This is another print statement")
```

Output:

```
Hello WorldThis is another print statement
```

Example 2:

```
print("Hello World", end="joker")
print("This is another print statement")
print("This is a good language", end="    ")
print("Goodbye")
```

Output:

```
Hello WorldjokerThis is another print statement
This is a good language    Goodbye
```

Escape Sequences

The programmers refer to the "backslash (\)" character as an escape character. In other words, it has a special meaning when we use it inside the strings. As the name suggests, the escape character escapes the characters in a string for a moment to introduce unique inclusion.

- An Escape Sequence character in Python is a sequence of characters that represents a single character.
- It does not represent itself when used inside string literal or character.
- It is composed of two or more characters starting with backslash \ but acts as a single character. Example \n depicts a new line character.

ESCAPE SEQUENCES	DESCRIPTION
\n	Inserts a new line in the text at the point
\\	Inserts a backslash character in the text at the point
\”	Inserts a double quote character in the text at that point
\’	Inserts a single quote character in the text at that point
\t	Inserts a Tab in the text at that point
\b	Inserts a backspace in the text at that point

Some examples:

Example 1

```
print("Hello this is python.\nIt is a good language")
```

Output

```
Hello this is python.
It is a good language
```

Example 2

```
print("Hello this is python.\tIt is a good language")
```

Output

```
Hello this is python.   It is a good language
```

Comments

Programming reflects your way of thinking in order to describe the single steps that you took to solve a problem using a computer. Commenting your code helps explain your thought process, and helps you and others to understand later the intention of your code. This allows you to find errors more easily, to fix them, to improve the code later on, and to reuse it in other applications as well.

Comments are used to write something which the programmer does not want to execute. Comments can be written to mark author name, date when the program is written, adding notes for your future self, etc.

Comments describe what is happening inside a program so that a person looking at the source code does not have difficulty figuring it out.

There are two types of comments in Python Language:

- Single line comment
- Multi line comment

Single line comment

Single Line comments are the comments which are written in a single line, i.e., they occupy the space of a single line.

We use # (hash/pound to write single-line comments).

Example

```
print("Hello Student.")
# I can write whatever I want to write
print("This is just a print statement.")  # I can write this
too
# This Is a comment and It will not execute
```

Output

```
Hello Student.
This is just a print statement.
```

Multi line comments

Multi-Line comments are the comments which are created by using multiple lines, i.e., they occupy more than one line in a program.

We use " " " ….. Comment …. " " " for writing multi-line comments in Python (Use lines enclosed with three quotes for writing multi-line comments). An example of a multi-line comment is shown below:

Example

```
"""This is a comment
Date: 2 January 1996
Multi-line comment ends here
"""
print("Hello Student!!!")
# Remember this is Single Line Comment
print("I am learning Python.")
```

Output

```
Hello Student!!!
I am learning Python.
```