

Executing, output and input statements

Monday, December 13, 2021 6:50 PM

RUNNING CODE VIA TERMINAL

In Python, code can be run through the command line by simply using the .py file extension. An example: `py file_name.py`. The following is a command line example:

```
py float.py
1.89
<class 'float'>
```

Comments can be added to Python code by using the #symbol before any text that comes after it. This is for single line comments. For multi-line comments, you can highlight any number of lines of code and press CTRL + /.

Additionally, python can be used within the interactive python console by typing `python` and pressing ENTER, this will open the python interpreter. To close the Python interpreter, type `quit()`.

OUTPUT

In Python, output is also simpler than in other high level programming languages. In C# for example the command for output is:

```
Console.WriteLine("Hello World");
```

Output is the method of displaying information on screen to the user. This is achieved by the `print()` method, built in and supported by Python. The `print()` method can accept and display many different data types via an operation known as concatenation but more on this later.

See the following example:

```
In [8]: print("You are learning Python")
```

```
Out[10]: 'You are learning Python'
```

In the `print()` method, the expressions are placed within the parenthesis. If there is more than one expression to be displayed through one print statement, they must be separated by commas. See the following example:

```
In [11]: print(1,2,3,4)
```

```
Out[10]: 1 2 3 4
```

Multiple different data types can be printed at the same time (see the concatenation section for more details), example:

```
In [12]: print(1,2,3,4, "Joe", True, 3.4)
```

```
Out[10]: 1 2 3 4 Joe True 3.4
```

The following shows a print statement syntax error if commas are not used:

File "h:\YEAR 3__pycache__\Basics_project\output.py", line 1

```
print(1 2 3 4 "Joe" True 3.4)
      ^
```

SyntaxError: invalid syntax

INPUT

Similarly, input in Python is as simple and more simple than other high level programming languages, input can be achieved by the `input()` method. See the following example which combines both the `print()` and `input()` methods:

```
In [13]: print("What is your name?")
         name1=input()
         print("Well met!", name1)
```

Out[10]: What is your name?

In [13]: Joe

Out[10]: Well met! Joe