

[1] Python Fundamentals

December 22, 2022

Python Fundamentals

Python Data Types

0.1 1.1 Numbers

Python has two basic number types, integer and float

For example, 2 is an integer and 2.0 is a floating point number which has a decimal attached to it. We can perform arithmetic operations to either of these numbers types. Let's try!

```
[1]: # addition  
5 + 2
```

```
[1]: 7
```

```
[2]: # Subtraction  
5 - 2
```

```
[2]: 3
```

```
[3]: # multiplication  
5 * 2
```

```
[3]: 10
```

With a forward slash, we can divide numbers in Python (note, two integers are giving floating point in results).

If you are using Python 2, you need to add 2.0 in the expression (e.g. 5 / 2.0) to get the same results.

```
[4]: 5/2
```

```
[4]: 2.5
```

We can compute the power of a number (exponent) with two Asterix ** together

```
[5]: 5 ** 2
```

```
[5]: 25
```

Python follow the order of arithmetic operations, for example: for $1 + 2 * 3 + 4$ Python will first multiply 2 & 3 then perform the other operations.

```
[6]: 1 + 2 * 3 + 4
```

```
[6]: 11
```

A good practice is to use parentheses “()” to tell the Python, which operation needs to be performed and to clarify the order. The operations in the parentheses will be performed first.

```
[7]: (1 + 2) * (3 + 4)
```

```
[7]: 21
```

Modulus or **mod** operation is a “%” (percentage sign) in Python. It returns what remains after a division. *With mod operation, we can check if the number is even or odd. Mod (%) returns 0, for even number and returns 1 for odd numbers.*

```
[8]: 6 % 2 # returns 0, 6 is even no.
```

```
[8]: 0
```

```
[9]: 5 % 2 # returns 1, 5 is odd no. (if we divide 5 by 2, remainder is 1)
```

```
[9]: 1
```

Variables: * Variables are containers for storing data values.

- Unlike other programming languages, Python has no command for declaring a variable.
- A variable is created the moment you first assign a value to it.

At many occasions, we pick a variable with some name and assign object or a data type to that variable. We can do this in Python using equal sign operator “=”. For example, if we assign a value of 5 to a variable “x”, whenever we call x, this will return 5 in the output.

It is a common practice to create a **variable with multiple names**. A good way is to separate them with underscore “_” between the words. By doing this, you will easily identify the variables e.g. total_profit, total_loss, first_name etc.

```
[10]: name_of_the_variable = 20
```

```
[11]: name_of_the_variable
```

```
[11]: 20
```

Important regarding variable name: Before we move on to the next data type, we should know that, in Python: * variable name can not be started with number (e.g. 1var, 2x etc) * variable name can not be started with special characters (e.g. !, @, #, !y, +, - etc) * variable name are case sensitive, “Name” is not the same as “name” * reserved words can not be a variable name, e.g. “class” is a reserve word in Python, it can not be used as a variable name, but “klass” and “Class” work fine.

```
[12]: # Let's try x = 5 and y = 3  
x = 5  
y = 3
```

```
[13]: # what is x now!  
x
```

```
[13]: 5
```

```
[14]: # what is y now!  
y
```

```
[14]: 3
```

We can perform arithmetic operations using these variable.

```
[15]: # Adding x and y  
x + y
```

```
[15]: 8
```

```
[16]: # Multiplying x and y  
x * y
```

```
[16]: 15
```

After performing any arithmetic operation, we can assign the result to a new variable.

```
[17]: # Subtraction of the variables and assigning the result to a new variable  
z = x - y      # 5 - 3
```

```
[18]: z
```

```
[18]: 2
```

We can re-assign a value to the same variables. This will replace the existing value to its new value.

```
[19]: x = y + 2  
x = x * x
```

```
[20]: x
```

```
[20]: 25
```

Let's try some invalid variable names!

```
[21]: 1var = 1
```

```
File "C:\Users\ahmad\AppData\Local\Temp\ipykernel_21648\1504440605.py", line  
1var = 1
```

```
^
SyntaxError: invalid syntax
```

```
[22]: $var = 3
```

```
File "C:\Users\ahmad\AppData\Local\Temp\ipykernel_21648\3251433108.py", line
    $var = 3
    ^
SyntaxError: invalid syntax
```

0.2 Booleans

Booleans are simply True and False with capital T and Capital F – just a customized version of 1 and 0.

```
[23]: True
```

```
[23]: True
```

```
[24]: False
```

```
[24]: False
```

```
[25]: 10 == 10
```

```
[25]: True
```

```
[26]: 5 == 10
```

```
[26]: False
```

```
[27]: a = 10
      b = 5
      a == b
```

```
[27]: False
```

0.3 Comparison Operators

`x == y` # x is equal to y

`x != y` # x is not equal to y

`x > y` # x is greater than y

`x < y` # x is less than y

`x >= y` # `x` is greater than or equal to `y`

`x <= y` # `x` is less than or equal to `y`

`x is y` # `x` is the same as `y`

`x is not y` # `x` is not the same as `y`

```
[28]: x = 10  
      y = 20
```

```
[29]: x == y # x is equal to y
```

```
[29]: False
```

```
[30]: x != y # x is not equal to y
```

```
[30]: True
```

```
[31]: x > y # x is greater than y
```

```
[31]: False
```

```
[32]: x < y # x is less than y
```

```
[32]: True
```

```
[33]: x >= y # x is greater than or equal to y
```

```
[33]: False
```

```
[34]: x <= y # x is less than or equal to y
```

```
[34]: True
```

```
[35]: x is y # x is the same as y
```

```
[35]: False
```

```
[36]: x is not y # x is not the same as y
```

```
[36]: True
```

0.4 Logical Operators

There are three logical operators: and, or, and not

```
[37]: x = 10  
      print (x > 0 and x < 10)
```

```
False
```

```
[38]: print (not (x%2 == 0))
```

False

```
[39]: print (10 or True) # 10
      print (10 and True) # True
      print (10 and False) # False
      print (10 or False) # 10
```

10

True

False

10

1 Exercise

1.1 Sales Tax

Write a program that will ask the user to enter the amount of a purchase. The program should then compute the state and county sales tax. Assume the state sales tax is 5 percent and the county sales tax is 2.5 percent. The program should display the following: * The amount of the purchase * The state sales tax * The county sales tax * The total sales tax * and the total of the sale (which is the sum of the amount of purchase plus the total sales tax).

1.2 Tip, Tax and Total

Write a program that calculates the total amount of a meal purchased at a restaurant. The program should ask the user to enter the charge for the food, and then calculate the amount of a 18 percent tip and 7 percent sales tax. Display each of these amounts and the total.

1.3 Ingredient Adjuster

A cookie recipe calls for the following ingredients: * 1.5 cups of sugar * 1 cup of butter * 2.75 cups of flour

The recipe produces 48 cookies with this amount of the ingredients. Write a program that asks the user how many cookies he or she wants to make, and then displays the number of cups of each ingredient needed for the specified number of cookies.

1.4 Male and Female Percentages

Write a program that asks the user for the number of males and the number of females registered in a class. The program should display the percentage of males and females in the class.

Hint: Suppose there are 8 males and 12 females in a class. There are 20 students in the class. The percentage of males can be calculated as $8 \div 20 = 0.4$, or 40%. The percentage of females can be calculated as $12 \div 20 = 0.6$, or 60%.

1.5 Stock Transation Program

Last month Joe purchased some stock in Acme Software, Inc. Here are the details of the purchase:

* The number of shares that Joe purchased was 2,000. * When Joe purchased the stock, he paid 40.00 Dollar per share. * Joe paid his stockbroker a commission that amounted to 3 percent of the amount he paid for the stock.

Two weeks later Joe sold the stock. Here are the details of the sale: * The number of shares that Joe sold was 2,000. * He sold the stock for 42.75 Dollar per share. * He paid his stockbroker another commission that amounted to 3 percent of the amount he received for the stock.

Write a program that displays the following information: * The amount of money Joe paid for the stock. * The amount of commission Joe paid his broker when he bought the stock. * The amount that Joe sold the stock for. * The amount of commission Joe paid his broker when he sold the stock. * Display the amount of money that Joe had left when he sold the stock and paid his broker (both times). If this amount is positive, then Joe made a profit. If the amount is negative, then Joe lost money.

[]: