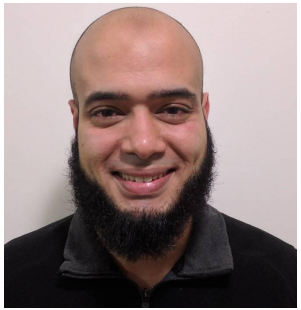
Python Programming

Arithmetic Operators

Mostafa S. Ibrahim
Teaching, Training and Coaching since more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)



Arithmetic Operators

```
2   x = 6
3   y = 3
4
5   # Binary Operator
6   print(x + y)            # 9
7   print(x + 2 * y - 1)    # 11
8
9   print(x / y)            # 2.0
10  z = (x + y) / 3 / 3
11  print(z)                # 1.0
12
13  # Unary Operator
14  print(-z)               # -1.0
```

- Arithmetic: traditional operations (e.g. addition) on numbers
- Operator: Symbol
  - Arithmetic Operators: + - * / // %
- x + 2 * y - 1
  - We call it **expression**
  - + * - are **operators**
  - x, 2, y, 1 are **operands**
- Parentheses (): applied first

Binary and Unary Operators

- **Binary** operators needs 2 operands
  - 3 + 5
  - 4 * 6
  - 5 - 2
- **Unary** Operator takes 1 operand only
  - -7
    - --7 = 7
    - ---7 = -7
  - +7
  - So + and - can be used as binary and unary operator

```
3   print(7)          # 7
4   print(+7)         # 7
5   print(+++7)       # 7
6
7   print(-7)         # -7
8   print(--7)        # 7
9   print(---7)       # -7
10  print(----7)      # 7
11
12  print(7 -- 5)       # 12
13  print(7 --- 5)      # 2
14
```

Division (/) and Floor Division (//)

- Division: results in a float
- Floor division also called Integer Division: results in integer

```
3   print(14 / 2)   # 7.0
4   print(14 / 3)   # 4.666666666666667
5   print(14 / 40)  # 0.35
6
7   print(14 // 2)   # 7
8   print(14 // 3)   # 4
9   print(14 // 40)  # 0
10
11  print(14 / 8)     # 1.75
12  print(14 // 8)    # 1   round down to a small value 1 = min(1, 2)
13  print(-14 // 8)   # -2  = -2 = min(-2, -1)
14
15  print(type(14 / 7))     # float
16  print(type(14 // 7))    # int
17
18
```

Division by power of 10s

```
2   num = 12345
3
4   print(num / 10)        #  1234.5
5   print(num / 100)       #  123.45
6   print(num / 1000)      #  12.345
7   print(num / 10000)     #  1.2345
8   print(num / 100000)    # 0.12345
9
10  # Remove last digits
11  print(num // 10)       # 1234
12  print(num // 100)      # 123
13  print(num // 1000)     # 12
14  print(num // 10000)    # 1
15  print(num // 100000)   # 0
16
```

- Dividing by 10 removes last digit
- Dividing by 100 remove last 2 digits and so on
- Notice num/1000 is same as num/10/10/10

Power Operator (**)

- Aka Exponentiation
- 5 ** 3 = 5^3 = 5 is being raised to the 3rd power
  - 5 * 5 * 5 = 5 multiplied in each self 3 times

```
4   print(2 ** 4)       # 16
5   print(2 ** -4)      # 0.0625 = 1/16
6
7   print(5 ** 0)       # 1
8   print(0 ** 5)       # 0
9
10  print(2.1 ** 4)     # 19.448100000000004
11  print(2 ** 4.1)     # 17.148375400580687
12
13  # Remove last K digits
14  num, k = 12345, 3
15  print(num // (10 ** k)) # 12
16
```

Modulus Operator (%)

- Returns the remainder rather than the quotient after division
- 6 % 2 = 0 (6 is divisible by 2)
- 6 % 4 = 2 (6 is not divisible by 4)
- We will study in details

Even and odd

- Even number is divisible by 2
  - E.g. 2, 4, 6, 8, 10, 12, …
  - 8/2 = 4 ⇒ Even
  - So always **number.0**
- Odd number is not divisible by 2
  - E.g. 1, 3, 5, 7, 11, …
  - Let’s divide them by 2
  - ½ = 0.5
  - 3/2 = 1.5
  - So 0.5 1.5 2.5 3.5 4.5 5.5
  - Like 0.5 + (1, 2, 3, 4, 5….)
  - So always **number.5**

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."