Basic Mathematical Operations

In [1]:

```
print(3 + 2)   # addition(+)
print(3 - 2)   # subtraction(-)
print(3 * 2)   # multiplication(*)
print(3 / 2)   # division(/)
print(3 ** 2)  # exponential(**)
print(3 % 2)   # modulus(%)
print(3 // 2)  # Floor division operator(//)
```

```
5
1
6
1.5
9
1
1
```

Checking data types

In [2]:

```
print(type(10))                  # Int
print(type(3.14))                # Float
print(type(1 + 3j))              # Complex
print(type('Sanket'))            # String
print(type([1, 2, 3]))           # List
print(type({'name':'Sanket'}))   # Dictionary
print(type({9.8, 3.14, 2.7}))    # Set
print(type((9.8, 3.14, 2.7)))    # Tuple
print(type(3 == 3))              # Bool
print(type(3 >= 3))              # Bool
```

```
<class 'int'>
<class 'float'>
<class 'complex'>
<class 'str'>
<class 'list'>
<class 'dict'>
<class 'set'>
<class 'tuple'>
<class 'bool'>
<class 'bool'>
```

Variables in Python

In [5]:

```
first_name = 'Sanket'
last_name = 'Mahajan'
country = 'India'
city = 'Jalgaon'
age = 21
is_married = False
skills = ['CV', 'ML', 'AI', 'SQL', 'Python']
person_info = {
    'firstname':'Sanket',
    'lastname':'Mahajan',
    'country':'India',
    'city':'Jalgaon'
    }
```

Printing the values stored in the variables

In [7]:

```
print('First name:', first_name)
print('First name length:', len(first_name))
print('Last name: ', last_name)
print('Last name length: ', len(last_name))
print('Country: ', country)
print('City: ', city)
print('Age: ', age)
print('Married: ', is_married)
print('Skills: ', skills)
print('Person information: ', person_info)
```

```
First name: Sanket
First name length: 6
Last name:  Mahajan
Last name length:  7
Country:  India
City:  Jalgaon
Age:  21
Married:  False
Skills:  ['CV', 'ML', 'AI', 'SQL', 'Python']
Person information:  {'firstname': 'Sanket', 'lastname': 'Mahajan', 'country': 'India', 'city': 'Jalgaon'}
```

Declaring multiple variables in one line

In [8]:

```
first_name, last_name, country, age, is_married = 'Sanket', 'Mahajan', 'India', 21, False

print(first_name, last_name, country, age, is_married)
print('First name:', first_name)
print('Last name: ', last_name)
print('Country: ', country)
print('Age: ', age)
print('Married: ', is_married)
```

```
Sanket Mahajan India 21 False
First name: Sanket
Last name:  Mahajan
Country:  India
Age:  21
Married:  False
```

Arithmetic Operations in Python

Integers

In [9]:

```
print('Addition: ', 1 + 2)
print('Subtraction: ', 2 - 1)
print('Multiplication: ', 2 * 3)
print ('Division: ', 4 / 2)                         # Division in python gives floating number
print('Division: ', 6 / 2)
print('Division: ', 7 / 2)
print('Division without the remainder: ', 7 // 2)   # gives without the floating number or without the remaining
print('Modulus: ', 3 % 2)                           # Gives the remainder
print ('Division without the remainder: ', 7 // 3)
print('Exponential: ', 3 ** 2)                     # it means 3 * 3
```

```
Addition:  3
Subtraction:  1
Multiplication:  6
Division:  2.0
Division:  3.0
Division:  3.5
Division without the remainder:  3
Modulus:  1
Division without the remainder:  2
Exponential:  9
```

Floating numbers

In [11]:

```
print('Floating Number,PI :', 3.14)
print('Floating Number, gravity :', 9.81)
```

```
Floating Number,PI : 3.14
Floating Number, gravity : 9.81
```

Complex numbers

In [12]:

```
print('Complex number: ', 1 + 1j)
print('Multiplying complex number: ',(1 + 1j) * (1-1j))
```

```
Complex number:  (1+1j)
Multiplying complex number:  (2+0j)
```

In [13]:

```
# Declaring the variable at the top first

a = 3 # a is a variable name and 3 is an integer data type
b = 2 # b is a variable name and 3 is an integer data type

# Arithmetic operations and assigning the result to a variable
total = a + b
diff = a - b
product = a * b
division = a / b
remainder = a % b
floor_division = a // b
exponential = a ** b
```

In [14]:

```
# I should have used sum instead of total but sum is a built-in function try to avoid overriding builtin functions
print(total) # if you don't label your print with some string, you never know from where is  the result is coming
print('a + b = ', total)
print('a - b = ', diff)
print('a * b = ', product)
print('a / b = ', division)
print('a % b = ', remainder)
print('a // b = ', floor_division)
print('a ** b = ', exponential)
```

```
5
a + b =  5
a - b =  1
a * b =  6
a / b =  1.5
a % b =  1
a // b =  1
a ** b =  9
```

In [15]:

```
# Declaring values and organizing them together
num_one = 3
num_two = 4
```

In [16]:

```
# Arithmetic operations
total = num_one + num_two
diff = num_two - num_one
product = num_one * num_two
```

```
div = num_two / num_two
remainder = num_two % num_one
```

In [17]:

```
# Printing values with label
print('total: ', total)
print('difference: ', diff)
print('product: ', product)
print('division: ', div)
print('remainder: ', remainder)
```

```
total:  7
difference:  1
product:  12
division:  1.0
remainder:  1
```

In [18]:

```
# Calculating area of a circle
radius = 10                                 # radius of a circle
area_of_circle = 3.14 * radius ** 2         # two * sign means exponent or power
print('Area of a circle:', area_of_circle)
```

```
Area of a circle: 314.0
```

In [19]:

```
# Calculating area of a rectangle
length = 10
width = 20
area_of_rectangle = length * width
print('Area of rectangle:', area_of_rectangle)
```

```
Area of rectangle: 200
```

In [20]:

```
# Calculating a weight of an object
mass = 75
gravity = 9.81
weight = mass * gravity
print(weight, 'N')
```

```
735.75 N
```

In [21]:

```
print(3 > 2)     # True, because 3 is greater than 2
print(3 >= 2)    # True, because 3 is greater than 2
print(3 < 2)     # False,  because 3 is greater than 2
print(2 < 3)     # True, because 2 is less than 3
print(2 <= 3)    # True, because 2 is less than 3
print(3 == 2)    # False, because 3 is not equal to 2
print(3 != 2)    # True, because 3 is not equal to 2
print(len('mango') == len('avocado'))  # False
print(len('mango') != len('avocado'))  # True
print(len('mango') < len('avocado'))   # True
print(len('milk') != len('meat'))      # False
print(len('milk') == len('meat'))      # True
print(len('tomato') == len('potato'))  # True
print(len('python') > len('dragon'))   # False
```

```
True
True
False
True
True
False
True
False
True
True
False
True
True
False
```

In [22]:

```
# Boolean comparison
print('True == True: ', True == True)
print('True == False: ', True == False)
print('False == False:', False == False)
print('True and True: ', True and True)
print('True or False:', True or False)
```

```
True == True:  True
True == False:  False
False == False: True
True and True:  True
True or False: True
```

In [23]:

```
# Another way comparison
print('1 is 1', 1 is 1)                   # True - because the data values are the same
print('1 is not 2', 1 is not 2)           # True - because 1 is not 2
print('S in Sanket', 'S' in 'Sanket')  # True - S found in the string
```

```python
print('T in Sanket', 'T' in 'Sanket') # False -there is no uppercase T
print('coding' in 'coding for all') # True - because coding for all has the word coding
print('a in an:', 'a' in 'an')      # True
print('4 is 2 ** 2:', 4 is 2 ** 2)   # True

print(3 > 2 and 4 > 3) # True - because both statements are true
print(3 > 2 and 4 < 3) # False - because the second statement is false
print(3 < 2 and 4 < 3) # False - because both statements are false
print(3 > 2 or 4 > 3)  # True - because both statements are true
print(3 > 2 or 4 < 3)  # True - because one of the statement is true
print(3 < 2 or 4 < 3)  # False - because both statements are false
print(not 3 > 2)     # False - because 3 > 2 is true, then not True gives False
print(not True)      # False - Negation, the not operator turns true to false
print(not False)     # True
print(not not True)  # True
print(not not False) # False
```

```
1 is 1 True
1 is not 2 True
S in Sanket True
T in Sanket False
True
a in an: True
4 is 2 ** 2: True
True
False
False
True
True
False
False
False
True
True
False
```

```
<>:2: SyntaxWarning: "is" with a literal. Did you mean "=="?
<>:3: SyntaxWarning: "is not" with a literal. Did you mean "!="?
<>:8: SyntaxWarning: "is" with a literal. Did you mean "=="?
<>:2: SyntaxWarning: "is" with a literal. Did you mean "=="?
<>:3: SyntaxWarning: "is not" with a literal. Did you mean "!="?
<>:8: SyntaxWarning: "is" with a literal. Did you mean "=="?
C:\Users\sanke\AppData\Local\Temp\ipykernel_19916\2819738200.py:2: SyntaxWarning: "is" with a literal. Did you mean
"=="?
  print('1 is 1', 1 is 1)                    # True - because the data values are the same
C:\Users\sanke\AppData\Local\Temp\ipykernel_19916\2819738200.py:3: SyntaxWarning: "is not" with a literal. Did you me
an "!="?
  print('1 is not 2', 1 is not 2)           # True - because 1 is not 2
C:\Users\sanke\AppData\Local\Temp\ipykernel_19916\2819738200.py:8: SyntaxWarning: "is" with a literal. Did you mean
"=="?
  print('4 is 2 ** 2:', 4 is 2 ** 2)   # True
```

```
In [ ]:
```