Arithmetic Operations in Python

Integers

In [2]:
```
print('Addition: ', 1 + 2)
```
```
Addition:  3
```

In [4]:
```
print('Subtraction: ', 2 - 1)
```
```
Subtraction:  1
```

In [6]:
```
print('Multiplication: ', 2 * 3)
```
```
Multiplication:  6
```

In [8]:
```
print ('Division: ', 4 / 2)            # Division in python gives floating number
```
```
Division:  2.0
```

In [10]:
```
print('Division: ', 6 / 2)
```
```
Division:  3.0
```

In [12]:
```
print('Division: ', 7 / 2)
```
```
Division:  3.5
```

In [14]:
```
print('Division without the remainder: ', 7 // 2)   # gives without the floating number or without the remainin
```
```
Division without the remainder:  3
```

In [16]:
```
print('Modulus: ', 3 % 2)                          # Gives the remainder
```
```
Modulus:  1
```

In [18]:
```
print ('Division without the remainder: ', 7 // 3)
```
```
Division without the remainder:  2
```

In [20]:
```
print('Exponential: ', 3 ** 2)                    # it means 3 * 3
```
```
Exponential:  9
```

Floating numbers

In [23]:
```
print('Floating Number,PI', 3.14)
```
```
Floating Number,PI 3.14
```

In [25]:
```
print('Floating Number, gravity', 9.81)
```
```
Floating Number, gravity 9.81
```

Complex numbers

In [28]:
```
print('Complex number: ', 1 + 1j)
```
```
Complex number:  (1+1j)
```

In [30]:
```
print('Multiplying complex number: ',(1 + 1j) * (1-1j))
```
```
Multiplying complex number:  (2+0j)
```

Declaring the variable at the top first

In [35]:
```
a = 3 # a is a variable name and 3 is an integer data type
b = 2 # b is a variable name and 2 is an integer data type
```

Arithmetic operations and assigning the result to a variable

In [38]:
```
total = a + b
diff = a - b
product = a * b
division = a / b
```

```
remainder = a % b
floor_division = a // b
exponential = a ** b
```

I should have used sum instead of total but sum is a built-in function try to avoid overriding builtin functions

In [41]:
```
print(total) # if you don't label your print with some string, you never know from where is  the result is comi
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
```

Declaring values and organizing them together

In [44]:
```
num_one = 3
num_two = 4
```

Arithmetic operations

In [47]:
```
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
```

Printing values with label

In [50]:
```
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
```

Calculating area of a circle

In [53]:
```
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
```

Calculating area of a rectangle

In [56]:
```
length = 10
width = 20
area_of_rectangle = length * width
print('Area of rectangle:', area_of_rectangle)
```

```
Area of rectangle: 200
```

Calculating a weight of an object

In [59]:
```
mass = 75
gravity = 9.81
weight = mass * gravity
print(weight, 'N')
```

```
735.75 N
```

In [61]:
```
print(3 > 2)     # True, because 3 is greater than 2
```

```
True
```

In [63]:
```
print(3 >= 2)    # True, because 3 is greater than 2
```

```
True
```

In [65]:
```
print(3 < 2)     # False,  because 3 is greater than 2
```

```
False
```

In [67]:
```
print(2 < 3)     # True, because 2 is less than 3
```

```
True
```

In [69]:
```
print(2 <= 3)    # True, because 2 is less than 3
```

```
True
```

In [71]:
```
print(3 == 2)    # False, because 3 is not equal to 2
```

```
False
```

In [73]:
```
print(3 != 2)    # True, because 3 is not equal to 2
```

```
True
```

In [75]:
```
print(len('mango') == len('avocado'))  # False
```

```
False
```

In [77]:
```
print(len('mango') != len('avocado'))  # True
```

```
True
```

In [79]:
```
print(len('mango') < len('avocado'))   # True
```

```
True
```

In [81]:
```
print(len('milk') != len('meat'))      # False
```

```
False
```

In [83]:
```
print(len('milk') == len('meat'))      # True
```

```
True
```

In [85]:
```
print(len('tomato') == len('potato'))  # True
```

```
True
```

In [87]:
```
print(len('python') > len('dragon'))   # False
```

```
False
```

Boolean comparison

In [90]:
```
print('True == True: ', True == True)
```

```
True == True:  True
```

In [92]:
```
print('True == False: ', True == False)
```

```
True == False:  False
```

In [94]:
```
print('False == False:', False == False)
```

```
False == False: True
```

In [96]:
```
print('True and True: ', True and True)
```

```
True and True:  True
```

In [98]:
```
print('True or False:', True or False)
```

```
True or False: True
```

Another way comparison

In [103…
```
print('1 is 1', 1 is 1)                    # True - because the data values are the same
```

```
1 is 1 True
<>:1: SyntaxWarning: "is" with 'int' literal. Did you mean "=="?
<>:1: SyntaxWarning: "is" with 'int' literal. Did you mean "=="?
C:\Users\kirti\AppData\Local\Temp\ipykernel_15328\2078387200.py:1: SyntaxWarning: "is" with 'int' literal. Did y
ou mean "=="?
  print('1 is 1', 1 is 1)                    # True - because the data values are the same
```

In [105…
```
print('1 is not 2', 1 is not 2)           # True - because 1 is not 2
```

```
1 is not 2 True
<>:1: SyntaxWarning: "is not" with 'int' literal. Did you mean "!="?
<>:1: SyntaxWarning: "is not" with 'int' literal. Did you mean "!="?
C:\Users\kirti\AppData\Local\Temp\ipykernel_15328\3437817676.py:1: SyntaxWarning: "is not" with 'int' literal. D
id you mean "!="?
  print('1 is not 2', 1 is not 2)           # True - because 1 is not 2
```

In [107…
```
print('A in Asabeneh', 'A' in 'Asabeneh') # True - A found in the string
```

```
A in Asabeneh True
```

In [109…
```
print('B in Asabeneh', 'B' in 'Asabeneh') # False -there is no uppercase B
```

```
B in Asabeneh False
```

In [111…
```
print('coding' in 'coding for all') # True - because coding for all has the word coding
```

```
True
```

In [113…
```
print('a in an:', 'a' in 'an')      # True
```

```
a in an: True
```

In [115…
```
print('4 is 2 ** 2:', 4 is 2 ** 2)   # True
```

```
4 is 2 ** 2: True
<>:1: SyntaxWarning: "is" with 'int' literal. Did you mean "=="?
<>:1: SyntaxWarning: "is" with 'int' literal. Did you mean "=="?
C:\Users\kirti\AppData\Local\Temp\ipykernel_15328\1007737545.py:1: SyntaxWarning: "is" with 'int' literal. Did y
ou mean "=="?
  print('4 is 2 ** 2:', 4 is 2 ** 2)   # True
```

In [117…
```
print(3 > 2 and 4 > 3) # True - because both statements are true
```

```
True
```

In [119…
```
print(3 > 2 and 4 < 3) # False - because the second statement is false
```

```
False
```

In [121…
```
print(3 < 2 and 4 < 3) # False - because both statements are false
```

```
False
```

In [123…
```
print(3 > 2 or 4 > 3)  # True - because both statements are true
```

```
True
```

In [125…
```
print(3 > 2 or 4 < 3)  # True - because one of the statement is true
```

```
True
```

In [127…
```
print(3 < 2 or 4 < 3)  # False - because both statements are false
```

```
False
```

In [129…
```
print(not 3 > 2)     # False - because 3 > 2 is true, then not True gives False
```

```
False
```

In [131…
```
print(not 3 > 2)     # False - because 3 > 2 is true, then not True gives False
```

```
False
```

In [133…
```
print(not True)      # False - Negation, the not operator turns true to false
```

```
False
```

In [135…
```
print(not False)     # True
```

```
True
```

In [137…
```
print(not not True)  # True
```

```
True
```

In [139…
```
print(not not False) # False
```

```
False
```

In [ ]: