Day3_ArithmeticOperations

May 19, 2025

1 Arithmetic Operations & Boolean Logic

Today, I explored how to perform arithmetic operations with integers, floats, and complex numbers in Python. I also practiced comparison operators and boolean logic. These concepts are essential for decision-making and numerical computations in any Python program.

2 Arithmetic Operations

3 Integers numbers

[2]:
```
# Integers

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
```

4 Floating numbers

[3]:
```
# Floating numbers
print('Floating Number,PI', 3.14)
print('Floating Number, gravity', 9.81)
```

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

5 Complex numbers

[4]:
```
# Complex numbers
print('Complex number: ', 1 + 1j)
print('Multiplying complex number: ',(1 + 1j) * (1-1j))
```

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

6 Declaring the variable at the top first

[5]:
```
# 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
```

7 Arithmetic operations and assigning the result to a variable

[6]:
```
# 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

# 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
```

8 Declaring values and organizing them together

[7]:
```
# Declaring values and organizing them together
num_one = 3
num_two = 4

# 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

# 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
```

9 Calculating area of a circle

[8]:
```
# 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
```

10 Calculating area of a rectangle

[9]:
```python
# 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
```

11 Calculating a weight of an object

[10]:
```python
# Calculating a weight of an object
mass = 75
gravity = 9.81
weight = mass * gravity
print(weight, 'N')
```

```
735.75 N
```

12 Boolean

[11]:
```python
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
```

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

[12]:
```python
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
```

```
False
True
```

```
True
False
True
True
False
```

13 Boolean comparison

[13]:
```
# 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
```

14 Another way comparison

[14]:
```
# 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('A in Asabeneh', 'A' in 'Asabeneh') # True - A found in the string
print('B in Asabeneh', 'B' in 'Asabeneh') # False -there is no uppercase B
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
```

```
1 is 1 True
1 is not 2 True
A in Asabeneh True
B in Asabeneh False
True
a in an: True
4 is 2 ** 2: True
```

```
<>:2: SyntaxWarning: "is" with 'int' literal. Did you mean "=="?
<>:3: SyntaxWarning: "is not" with 'int' literal. Did you mean "!="?
<>:8: SyntaxWarning: "is" with 'int' literal. Did you mean "=="?
<>:2: SyntaxWarning: "is" with 'int' literal. Did you mean "=="?
<>:3: SyntaxWarning: "is not" with 'int' literal. Did you mean "!="?
```

```
<>:8: SyntaxWarning: "is" with 'int' literal. Did you mean "=="?
C:\Users\aksha\AppData\Local\Temp\ipykernel_13152\4207222253.py:2:
SyntaxWarning: "is" with 'int' literal. Did you mean "=="?
  print('1 is 1', 1 is 1)                   # True - because the data values are
the same
C:\Users\aksha\AppData\Local\Temp\ipykernel_13152\4207222253.py:3:
SyntaxWarning: "is not" with 'int' literal. Did you mean "!="?
  print('1 is not 2', 1 is not 2)           # True - because 1 is not 2
C:\Users\aksha\AppData\Local\Temp\ipykernel_13152\4207222253.py:8:
SyntaxWarning: "is" with 'int' literal. Did you mean "=="?
  print('4 is 2 ** 2:', 4 is 2 ** 2)   # True
```

[15]:
```python
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
```

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

[ ]: