30 Days Of Python: Day 3 - Operators

LinkedIn | Follow @asabeneh

Author: Asabeneh Yetayeh

Second Edition: July, 2021

<< Day 2 | Day 4 >>

📘 Day 3

Boolean

A boolean data type represents one of the two values: *True* or *False*. The use of these data types will be clear once we start using the comparison operator. The first letter **T** for True and **F** for False should be capital unlike JavaScript. **Example: Boolean Values**

```
print(True)
print(False)
```

Operators

Python language supports several types of operators. In this section, we will focus on few of them.

Assignment Operators

Assignment operators are used to assign values to variables. Let us take = as an example. Equal sign in mathematics shows that two values are equal, however in Python it means we are storing a value in a certain variable and we call it assignment or a assigning value to a variable. The table below shows the different types of python assignment operators, taken from w3school.

Operator	Example	Same As
=	x = 5	x = 5
+=	x += 3	x = x + 3
-=	x -= 3	x = x - 3
*=	x *= 3	x = x * 3
/=	x /= 3	x = x / 3
%=	x %= 3	x = x % 3
//=	x //= 3	x = x // 3
**=	x **= 3	x = x ** 3
&=	x &= 3	x = x & 3
|=	x |= 3	x = x | 3
^=	x ^= 3	x = x ^ 3
>>=	x >>= 3	x = x >> 3
<<=	x <<= 3	x = x << 3

Arithmetic Operators:

- Addition(+): a + b
- Subtraction(-): a - b
- Multiplication(*): a * b
- Division(/): a / b
- Modulus(%): a % b
- Floor division(//): a // b
- Exponentiation(**): a ** b

Operator	Name	Example
+	Addition	x + y
-	Subtraction	x - y
*	Multiplication	x * y
/	Division	x / y
%	Modulus	x % y
**	Exponentiation	x ** y
//	Floor division	x // y

Example:Integers

```
# Arithmetic Operations in Python
# Integers

print('Addition: ', 1 + 2)        # 3
print('Subtraction: ', 2 - 1)     # 1
print('Multiplication: ', 2 * 3)  # 6
print ('Division: ', 4 / 2)       # 2.0  Division in Python gives floating number
print('Division: ', 6 / 2)        # 3.0
print('Division: ', 7 / 2)        # 3.5
print('Division without the remainder: ', 7 // 2)   # 3,  gives without the floating number or without the remaining
print ('Division without the remainder: ',7 // 3)   # 2
print('Modulus: ', 3 % 2)         # 1, Gives the remainder
print('Exponentiation: ', 2 ** 3) # 9 it means 2 * 2 * 2
```

Example:Floats

```
# Floating numbers
print('Floating Point Number, PI', 3.14)
print('Floating Point Number, gravity', 9.81)
```

Example:Complex numbers

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

Let's declare a variable and assign a number data type. I am going to use single character variable but remember do not develop a habit of declaring such types of variables. Variable names should be all the time mnemonic.

Example:

```
# 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

# I should have used sum instead of total but sum is a built-in function - try to avoid overriding built-in functions
print(total) # if you do not label your print with some string, you never know where the result is coming from
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 = ', exponentiation)
```

Example:

```
print('== Addition, Subtraction, Multiplication, Division, Modulus ==')

# 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_one
remainder = num_two % num_one

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

Let us start start connecting the dots and start making use of what we already know to calculate (area, volume,density, weight, perimeter, distance, force).

Example:

```
# 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)

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

# Calculating a weight of an object
mass = 75
gravity = 9.81
weight = mass * gravity
print(weight, 'N')                         # Adding unit to the weight

# Calculate the density of a liquid
mass = 75 # in Kg
volume = 0.075 # in cubic meter
density = mass / volume # 1000 Kg/m^3
```

Comparison Operators

In programming we compare values, we use comparison operators to compare two values. We check if a value is greater or less or equal to other value. The following table shows Python comparison operators which was taken from w3shool.

Operator	Name	Example
==	Equal	x == y
!=	Not equal	x != y
>	Greater than	x > y
<	Less than	x < y
>=	Greater than or equal to	x >= y
<=	Less than or equal to	x <= y

Example: Comparison Operators

```
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


# Comparing something gives either a True or False

print('True == True: ', True == True)
print('True == False: ', True == False)
print('False == False:', False == False)
```

In addition to the above comparison operator Python uses:

- *is*: Returns true if both variables are the same object(x is y)
- *is not*: Returns true if both variables are not the same object(x is not y)
- *in*: Returns True if the queried list contains a certain item(x in y)
- *not in*: Returns True if the queried list doesn't have a certain item(x in y)

```
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
```

Logical Operators

Unlike other programming languages python uses keywords *and*, *or* and *not* for logical operators. Logical operators are used to combine conditional statements:

Operator	Description	Example
and	Returns True if both statements are true	x < 5 and x < 10
or	Returns True if one of the statements is true	x < 5 or x < 4
not	Reverse the result, returns False if the result is true	not(x < 5 and x < 10)

```
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('True and True: ', True and True)
print(3 > 2 or 4 > 3)  # True - because both statements are true
print(3 > 2 or 4 < 3)  # True - because one of the statements is true
print(3 < 2 or 4 < 3)  # False - because both statements are false
print('True or False:', True or 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
```

🌕 You have boundless energy. You have just completed day 3 challenges and you are three steps ahead on your way to greatness. Now do some exercises for your brain and your muscles.

💻 Exercises - Day 3

1. Declare your age as integer variable
2. Declare your height as a float variable
3. Declare a variable that store a complex number
4. Write a script that prompts the user to enter base and height of the triangle and calculate an area of this triangle (area = 0.5 x b x h).

```
    Enter base: 20
    Enter height: 10
    The area of the triangle is 100
```

5. Write a script that prompts the user to enter side a, side b, and side c of the triangle. Calculate the perimeter of the triangle (perimeter = a + b + c).

```
Enter side a: 5
Enter side b: 4
Enter side c: 3
The perimeter of the triangle is 12
```

6. Get length and width of a rectangle using prompt. Calculate its area (area = length x width) and perimeter (perimeter = 2 x (length + width))
7. Get radius of a circle using prompt. Calculate the area (area = pi x r x r) and circumference (c = 2 x pi x r) where pi = 3.14.
8. Calculate the slope, x-intercept and y-intercept of y = 2x -2
9. Slope is (m = y2-y1/x2-x1). Find the slope and Euclidean distance between point (2, 2) and point (6,10)
10. Compare the slopes in tasks 8 and 9.
11. Calculate the value of y (y = x^2 + 6x + 9). Try to use different x values and figure out at what x value y is going to be 0.
12. Find the length of 'python' and 'dragon' and make a falsy comparison statement.
13. Use *and* operator to check if 'on' is found in both 'python' and 'dragon'
14. *I hope this course is not full of jargon*. Use *in* operator to check if *jargon* is in the sentence.
15. There is no 'on' in both dragon and python
16. Find the length of the text *python* and convert the value to float and convert it to string
17. Even numbers are divisible by 2 and the remainder is zero. How do you check if a number is even or not using python?
18. Check if the floor division of 7 by 3 is equal to the int converted value of 2.7.
19. Check if type of '10' is equal to type of 10
20. Check if int('9.8') is equal to 10
21. Writ a script that prompts the user to enter hours and rate per hour. Calculate pay of the person?

```
Enter hours: 40
Enter rate per hour: 28
Your weekly earning is 1120
```

22. Write a script that prompts the user to enter number of years. Calculate the number of seconds a person can live. Assume a person can live hundred years

```
Enter number of years you have lived: 100
You have lived for 3153600000 seconds.
```

23. Write a Python script that displays the following table

```
1 1 1 1 1
2 1 2 4 8
3 1 3 9 27
4 1 4 16 64
5 1 5 25 125
```

🎉 CONGRATULATIONS ! 🎉

<< Day 2 | Day 4 >>