Python Programming Language

Lecture 4: Operator, String and Problem Solving

Hafsa Sultana

Computer Science and Engineering

Khulna University

Today’s Learning Objectives

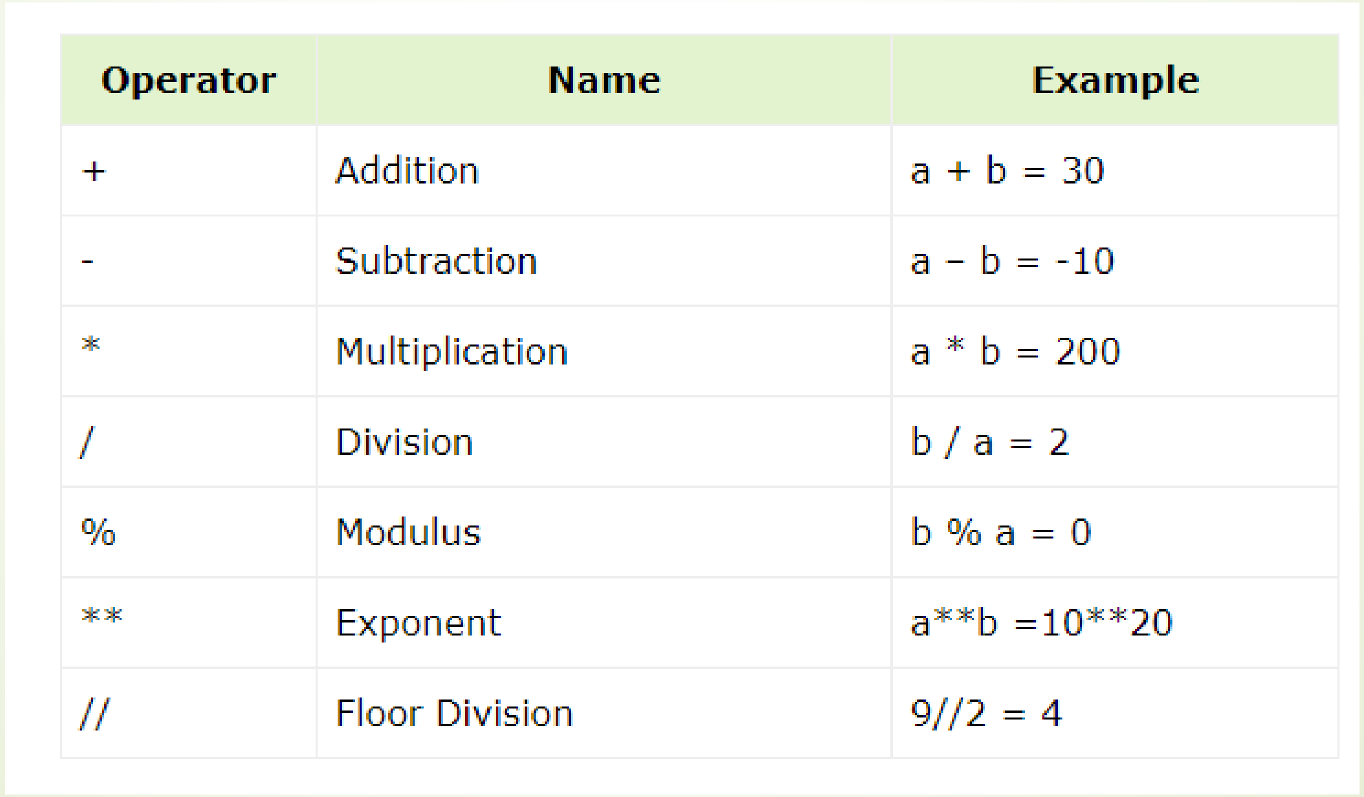
- Types of Operators
- Exercise of Operators
- String
- String Slicing
- Check String
- String Functions/ Methods
- Practical Exercises,

Python Operators

Operators are used to perform operations on variables and values.

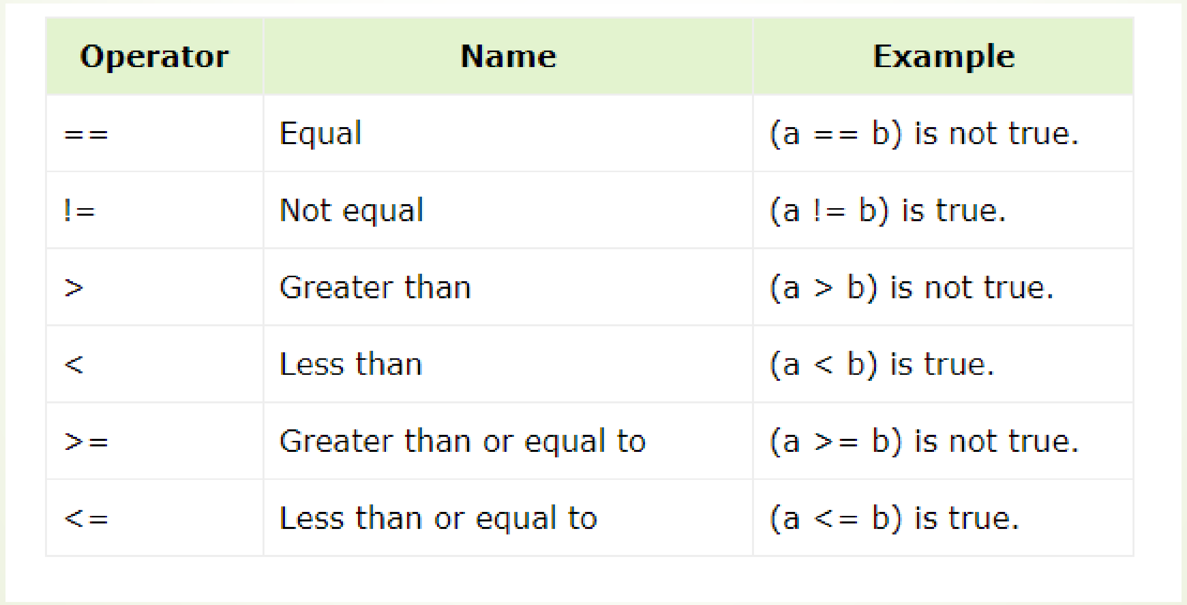
- Examples: +, -, *, /, ==, !=
- **Types of Operators:**
  - Arithmetic Operators
  - Comparison (Relational) Operators
  - Assignment Operators
  - Logical Operators
  - Bitwise Operators
  - Membership Operators
  - Identity Operators

1.Arithmetic Operators:



Operator	Name	Example
+	Addition	a + b = 30
-	Subtraction	a – b = -10
*	Multiplication	a * b = 200
/	Division	b / a = 2
%	Modulus	b % a = 0
**	Exponent	a**b =10**20
//	Floor Division	9//2 = 4

2. Comparison Operators:

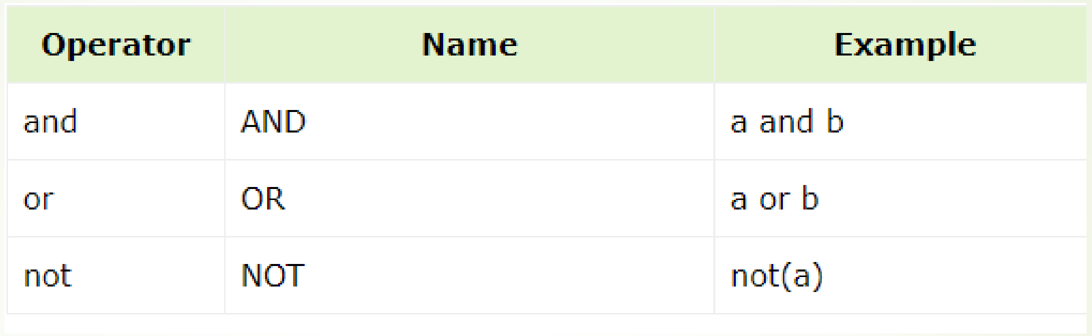


Operator	Name	Example
==	Equal	(a == b) is not true.
!=	Not equal	(a != b) is true.
>	Greater than	(a > b) is not true.
<	Less than	(a < b) is true.
>=	Greater than or equal to	(a >= b) is not true.
<=	Less than or equal to	(a <= b) is true.

3. Assignment Operators:

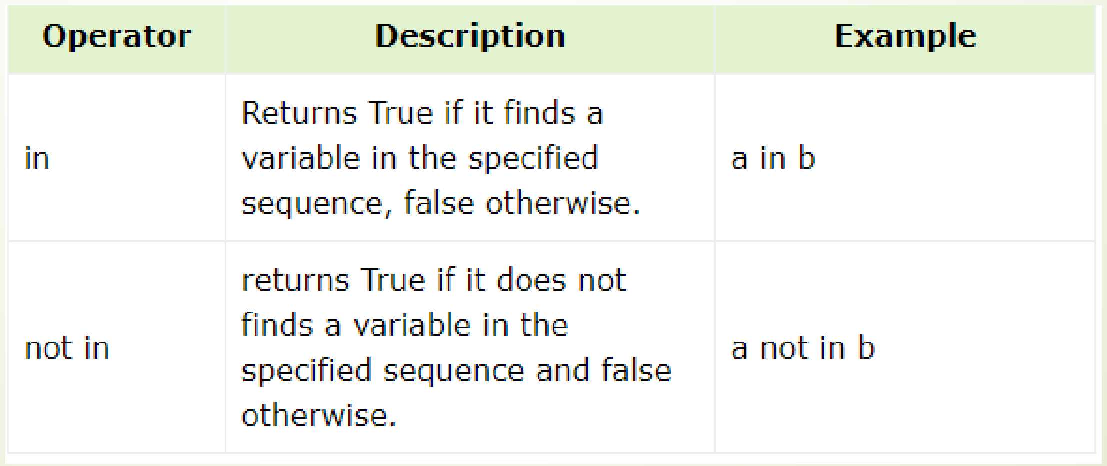
Operator	Example	Same As
=	a = 10	a = 10
+=	a += 30	a = a + 30
-=	a -= 15	a = a - 15
*=	a *= 10	a = a * 10
/=	a /= 5	a = a / 5
%=	a %= 5	a = a % 5
**=	a **= 4	a = a ** 4
//=	a //= 5	a = a // 5
&=	a &= 5	a = a & 5
|=	a |= 5	a = a | 5
^=	a ^= 5	a = a ^ 5
>>=	a >>= 5	a = a >> 5
<<=	a <<= 5	a = a << 5

4. Logical Operators:



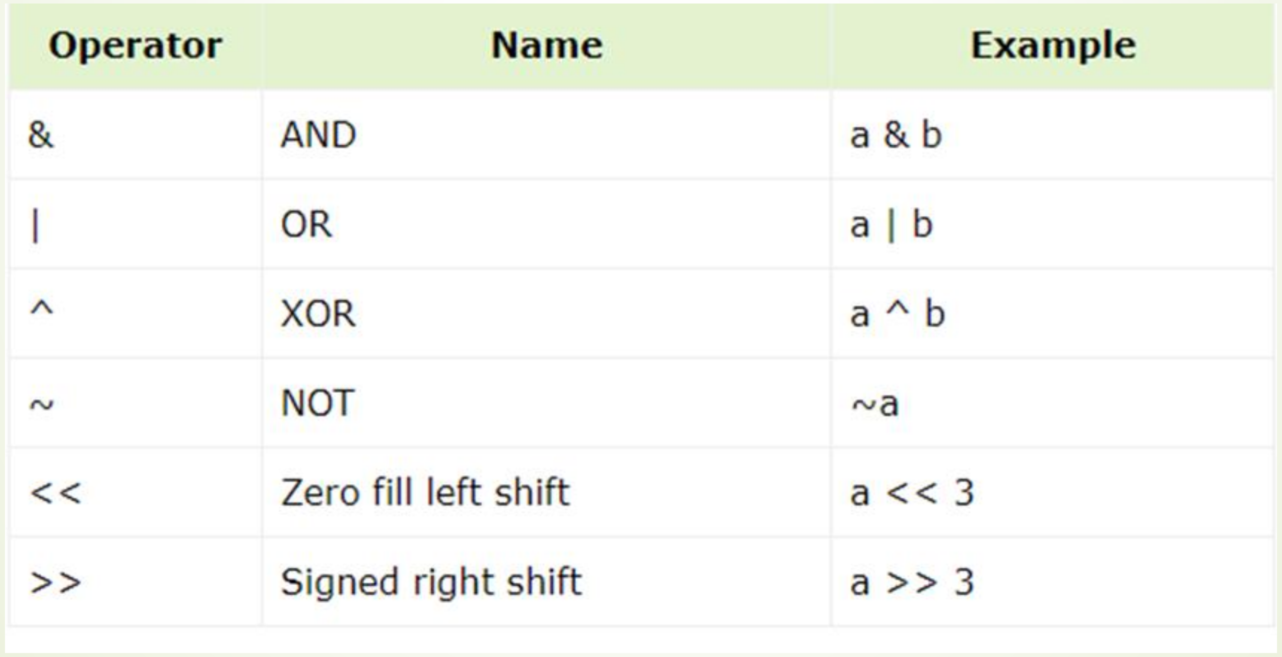
Operator	Name	Example
and	AND	a and b
or	OR	a or b
not	NOT	not(a)

5. Membership Operators:



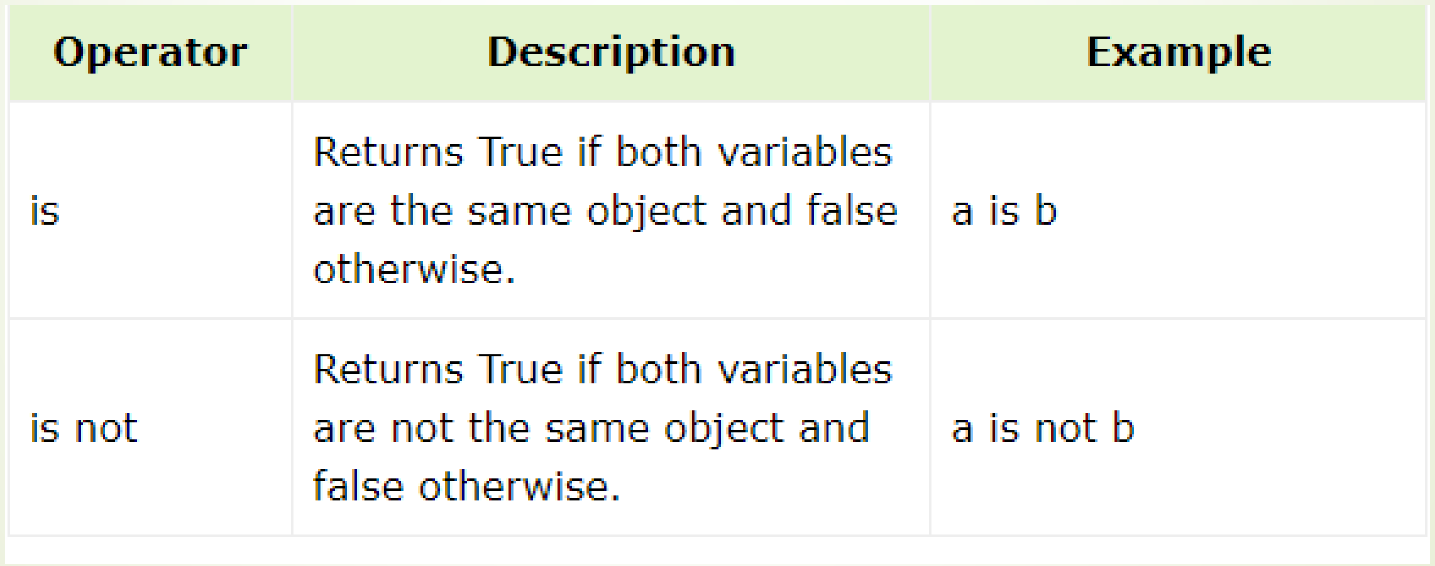
Operator	Description	Example
in	Returns True if it finds a variable in the specified sequence, false otherwise.	a in b
not in	returns True if it does not finds a variable in the specified sequence and false otherwise.	a not in b

6. Bitwise Operators:



Operator	Name	Example
&	AND	a & b
|	OR	a | b
^	XOR	a ^ b
~	NOT	~a
<<	Zero fill left shift	a << 3
>>	Signed right shift	a >> 3

7. Identity Operators:



Operator	Description	Example
is	Returns True if both variables are the same object and false otherwise.	a is b
is not	Returns True if both variables are not the same object and false otherwise.	a is not b

Operator

Precedence:

- **Example1:**

```
print((6 + 3) - (6 + 3))
print(100 + 5 * 3)
print(5 + 4 - 7 + 3)
```

- **Example2:** #Example of Python Membership Operator

```
a = 10
b = 20
c=b/a
list1 = [1, 2, 3, 4, 5 ]
print ("a: ", a, "b: ", b, "c: ", c, "list1:", list1)
```

```
print( a in list1)
print( b not in list1)
print( c in list1)
```

String

str = 'Hello World!’

- print (str)   # Hello World!
- print (str[0])   # H
- print (str[2:5])   #llo
- print (str[2:])   # llo World!
- print (str * 2)   # Hello World!Hello World!
- print (str + "TEST") # Hello World!TEST

String

Accessing Characters:

- **Example1:**

```
message = "Khulna University"
first_letter = message[0] # 'K'
last_letter = message[-1] # 'y'
print(first_letter, last_letter)

Sub_str = message[-4:-1]
print(Sub_str )    # sit
```

- **Example2:**

```
S1 = message[-10] # 'U'
S2 = message[-17] # 'K'
print(S1, S2)

or

length=len(message)
print(length)
S2 = message[- length] # 'y'
print("S2 is:  ",S2)
```

String Slicing

Extract substrings using colon (:) notation within square brackets.

Syntax: [start:end:step].

- **Example1:**

```
message = "Python_Class"
substring = message[1:6] # "ython" (extracts characters from index 1 to 6)
print(substring)
```

- **Example2:**

```
substring = message[1:8:2] # "yhnC" (extracts characters from index 1 to 8)
print(substring)
```

Check String

- **Example 1:**

```
txt = "Smiling is a kind act and the Sunnah of our beloved Prophet
      Muhammad (PBUH)."
print(" Sunnah " in txt)    #True
```

- **Example 2:**

```
print("expensive" not in txt)
print("Sunnah" not in txt)

find_txt = "expensive" not in txt
print("Is expensive in txt ? = ",find_txt)
```

String Function/ Methods

Strings provide various built-in methods for manipulation:

- **upper():** Convert to uppercase (Ex: message.upper()).
- **lower():** Convert to lowercase (Ex: message.lower()).
- **strip():** Remove leading and trailing whitespaces (Ex: message.strip()).
- **find(substring):** Find the index of the first occurrence of a substring (returns -1 if not found).
- **replace(old, new):** Replace all occurrences of a substring with another substring.
- **count():** Returns the number of times a specified value occurs in a string
- Many more! (Refer to Python documentation for the full list)

String Function

```
my_str = "  Helping others is a Sunnah  "

#upper():
print(my_str.upper())
#lower():
print(my_str.lower())
#title():
print(my_str.title())
#strip():
print(my_str.strip(" "))
```

```
#find():
print(my_str.find("Sunnah"))    #22
print(my_str.find("Helping"))   #2
#replace():
print(my_str.replace("Helping","Supporting"))
              #Supporting others is Sunnah.
#count()
print(my_str.count("i"))   #2
```

Practical Exercises

1. Reverse a String

   Input: Hello

   Output: olleH

2. Check if a String is a Palindrome

   Input: Enter the String: "racecar“

   Output: True

Practical Exercises

3. Count the Number of Vowels in a String

    Input: Enter the String: Hello

    Enter the count vowel: e

    Output: 1

4. Split a String by a Delimiter

    Input: Enter a string: "apple,banana,orange“

    Output: ["apple", "banana", "orange"] #input_string.split(',')

Practical Exercises

5. You have a dataset with strings. Each string contains:

- A sentence where the last character indicates the currency type (e.g., ‘$' for dollars, 'E' for euros).
- The 5th, 6th, and 7th characters represent the amount of money.

Your task is to extract the amount of money and the currency type from each string.

Practical Exercises

Missing char

Given a non-empty string and an int n, return a new string where the char at index n has been removed. The value of n will be a valid index of a char in the original string (i.e. n will be in the range 0..len(str)-1 inclusive).

missing_char('kitten', 1) → 'ktten'

missing_char('kitten', 0) → 'itten'

missing_char('kitten', 4) → 'kittn'

Practical Exercises

Make the sequence : abba

Given two strings, a and b, return the result of putting them together in the order abba, e.g. "Hi" and "Bye" returns "HiByeByeHi".

make_abba('Hi', 'Bye') → 'HiByeByeHi'

make_abba('Yo', 'Alice') → 'YoAliceAliceYo'

make_abba('What', 'Up') → 'WhatUpUpWhat'

Practical Exercises

7 Sln:

```
print("*")
print("**")
print("***")
print("****")
print("*****")
```

8 Sln:

```
print("    *    ")
print("   ***   ")
print("  *****  ")
print(" ******* ")
print("*********")
```

9 Sln:

```
print("    *    ")
print("   ***   ")
print("  *****  ")
print(" ******* ")
print("*********")
print(" ******* ")
print("  *****  ")
print("   ***   ")
print("    *    ")
```

10 Sln:

```
print("*****")
print("*   *")
print("*   *")
print("*   *")
print("*****")
```

11. Sln:

```
print("*   *")
print(" * * ")
print("  *  ")
print(" * * ")
print("*   *")
```

12 Sln:

```
print("    *    ")
print("   * *   ")
print("  *   *  ")
print(" *     * ")
print("*       *")
print(" *     * ")
print("  *   *  ")
print("   * *   ")
print("    *    ")
```

13. Sln:

```
print("*     *     *     *")
print(" *   * *   * *   * ")
print("  * *   * *   * *  ")
print("   *     *     *   ")
```

Debugging Tips

Use print statements to debug.

Check variable values and types.

Any Question?

References

- https://www.w3schools.com/python/
- https://www.tutorialspoint.com/python/