

## Shortcut operators

It's time for the next set of operators that make a developer's life easier.

Very often, we want to use one and the same variable both to the right and left sides of the `=` operator.

For example, if we need to calculate a series of successive values of powers of 2, we may use a piece like this:

```
x = x * 2
```

You may use an expression like this if you can't fall asleep and you're trying to deal with it using some good, old-fashioned methods:

```
sheep = sheep + 1
```

Python offers you a shortened way of writing operations like these, which can be coded as follows:

```
x *= 2
sheep += 1
```

1 Start coding or [generate](#) with AI.

Let's try to present a general description for these operations.

If `op` is a two-argument operator (this is a very important condition) and the operator is used in the following context:

```
variable = variable op expression
```

It can be simplified and shown as follows:

```
variable op= expression
```

Take a look at the examples below. Make sure you understand them all.

```
i = i + 2 * j ⇒ i += 2 * j
var = var / 2 ⇒ var /= 2
rem = rem % 10 ⇒ rem %= 10
j = j - (i + var + rem) ⇒ j -= (i + var + rem)
x = x ** 2 ⇒ x **= 2
```

## ✓ Comparison Operators

[Python Comparison Operators- W3School](#)

```
1 #experimenting with comparison operators
2 #remember that thr outcome of comparison is always a boolean (false or true) value
3
4 a = 10
5 b = 20
6
7 print(f"Does a equals b:\t {a == b}")
8 print(f"Does a not equals b: \t {a != b}")
9 print(f"Is a greater than b: \t {a > b}")
10 print(f"Is a less than b: \t {a < b}")
11 print(f"Is a greater than or equal to b: \t {a >= b}")
12 print(f"Is a less than or equal to b: \t {a <= b}")
```

```
↔ Does a equals b:           False
  Does a not equals b:      True
  Is a greater than b:      False
  Is a less than b:         True
  Is a greater than or equal to b:      False
  Is a less than or equal to b:      True
```

```
1 print(2==3)
```

```
False
```

```
1 print(2==2)
```

```
True
```

```
1 my_number = 60
2 print(my_number==60)
3 print(my_number==62)
```

```
True
False
```

```
1 print(my_number>60)
```

```
False
```

```
1 print(my_number>59)
```

```
True
```

```
1 print(my_number<96)
```

```
True
```

```
1 print(my_number>=60)
```

```
True
```

```
1 print(my_number<=60)
```

```
True
```

```
1 temperature = 19
2 print(temperature>20)
```

```
False
```

#### ✓ not equal operator

```
1 print(2 != 3)
```

```
True
```

```
1 #unlike sql python doesnt accept <>
2 print(2 <> 3)
```

```
File "<ipython-input-22-39815bba0ad4>", line 2
    print(2 <> 3)
           ^
SyntaxError: invalid syntax
```

```
1 print(temperature != 20)
```

```
True
```

```
1 print(temperature != 19)
```

```
False
```

#### ✓ Exercise: Take two number from a user, compare and return result.

```
1 number1 = float(input("Please enter your first number:"))
2 number2 = float(input("Please enter your second number:"))
3 print("is ", number1, "greater than", number2, number1>number2)
4 print(number1, "is less than", number2, number1<number2)
5 print(number1, "is equal to", number2, number1==number2)
```

```
Please enter your first number:25
Please enter your second number:5
```

```

is 25.0 greater than 5.0 True
25.0 is less than 5.0 False
25.0 is equal to 5.0 False

```

```

1 num = 300
2 if num > 0:
3     print(num, "is a positive number")
4 elif num < 0:
5     print(num, "is a negative number")
6 else:
7     print(num, "is zero")

```

↻ 300 is a positive number

```

1 # prompt: show an example of comparison operator using
2
3 x = 10
4 y = 5
5
6 print(x > y) # Output: True
7 print(x != y) #output: True
8 print(x < y) #output: False
9 print(x <= y) #output: False
10 print(x == y) #output: False

```

↻ True  
True  
False  
False  
False

```

1 # show an example of comparison operator using >
2 x = 10
3 y = 5
4
5 print(x > y) # Output: True
6

```

↻ True

## ✓ Over to you - Exercise - Lab

Using one of the comparison operators in Python, write a simple two-line program that takes the parameter *n* as input, which is an integer, and prints False if *n* is less than 100, and True if *n* is greater than or equal to 100.

```

1 # prompt: show an example of comparison operator using
2
3 x = 10
4 y = 5
5
6 print(x < y) # Output: False

```

```

1 #two line program
2
3 n = int(input("Enter a number: "))
4 print(n >= 100)
5
6
7

```

↻ Enter a number: 500  
True

## ✓ Solution

```

1 n = int(input("Enter a number: "))
2 print(n >= 100)

```

↻ Enter a number: 100  
True

## Algorithm

[Algorithm BBC](#)

## ✓ If Statement

[Python Flow Control - If Statements](#)

You can also watch the videos and Follow along.

if statements in python allows the programme to run a piece of code based on the outcome of the conditions. Conditions are usually the comparison operators.

In real life, we make these kind of decisions all the time such as:

```
if the temperature is hot, turn on the fan.
if it is raining, take umbrella.
```

```
1 #using if, lets check whether a number is positive or not?
2 num = int(input("Enter a number: "))
3
4 if num > 0:
5     print(num, "is a number")
```

```
↩ Enter a number: 7
7 is a number
```

```
1 #in the previous example, it doesn't account what happens if the number wasn't positional (if the condition is false)
2 num = float(input("Enter a number: "))
3
4 if num > 0:
5     print(num, "thats a positive number")
6 else:
7     print("thats a negative number")
```

```
↩ Enter a number: -85
thats a negative number
```

```
1 #elif is for than one condition
2 bird = str(input("what animal is that? "))
3
4 if bird == "parrot":
5     print("this", bird, "is bereft of life")
6 elif bird == "human":
7     print(bird, "\b, the most deadliest of animals")
8 else:
9     print(bird, "\b? that's a nice", bird)
```

```
↩ what animal is that? cat
cat? that's a nice cat
```

```
1 temp = int(input("What is the temperature outside? "))
2
3 if temp > 25:
4     print("Cor, it's hot out today")
5 elif 18 <= temp <= 25:
6     print("Cor, it's nice out today")
7 else:
8     print("Freezing innit?!")
```

```
↩ What is the temperature outside? 27
Cor, it's hot out today
```

## ✓ Follow along exercises for you to practice

```
1 #21a for you to complete
2 annualSales = 600000
3 if annualSales >= 500000:
4     # add print statements here
5     print("God Customer")
6     print("Thank you for your business")
```

```
↩ God Customer
Thank you for your business
```

```

1 #21b for you to complete, please add the two elif statements as you saw in the video
2 annualSales = 300000
3 if annualSales >= 500000:
4     print("Gold Customer")
5 print("Thank you for your business")

```

Thank you for your business

```

1 #21c Else for you to complete
2 annualSales = 300000
3 if annualSales >= 500000:
4     print("Gold Customer")
5 elif annualSales >= 300000:
6     print("Silver Customer")
7 elif annualSales >= 100000:
8     print("Bronze Customer")
9 print("Thank you for your business")

```

Silver Customer  
Thank you for your business

```

1 annualseales = int(input("What is your annual sales? "))
2 region = str(input("Which region are you in? "))
3
4 if annualseales >= 500000:
5     print("Platinum Customer")
6 elif annualseales >= 300000:
7     print("Gold Customer")
8     if region == "East":
9         print("Send a paddle")
10    elif region == "West":
11        print("Send a tuna")
12    else:
13        print("Send a ring")
14 elif annualseales >= 100000:
15     print("Silver Customer")
16 else:
17     print("Bronze Customer")
18     print("Think about upgrading")
19 print("Thank you for your business")

```

What is your annual sales? 20  
Which region are you in? West  
Bronze Customer  
Think about upgrading  
Thank you for your business

```

1 #21d Nested Statement for you to practice and follow along
2 annualSales = 300000
3 region = "North"
4
5 if annualSales >= 500000:
6     print("Gold Customer")
7 elif annualSales >= 300000:
8     print("Silver Customer")
9 #add if statement here
10
11
12
13 elif annualSales >= 100000:
14     print("Bronze Customer")
15 print("Thank you for your business")

```

```

1 #21d Compund Statement for you to practice and follow along
2 annualSales = 300000
3 newCustomer = False
4
5 if annualSales >= 500000:
6     print("Gold Customer")
7 elif annualSales >= 300000:
8     print("Silver Customer")
9 #add compound conditional statement here
10
11 elif annualSales >= 100000:
12     print("Bronze Customer")
13 print("Thank you for your business")

```

More If statements for you to practice below

```

1 number1 = float(input("Please enter a number:"))
2 number2 = float(input("Please enter a number:"))
3
4 if number2>number1:
5     print(number2, "is greater than", number1)

```

```

1 number1 = float(input("Please enter a number:"))
2 number2 = float(input("Please enter a number:"))
3
4 if number2>number1:
5     print(number2, "is greater than", number1)
6 else:
7     print(number2, "is not greater than", number1)
8

```

```

1 # Read two numbers
2 number1 = int(input("Enter the first number: "))
3 number2 = int(input("Enter the second number: "))
4
5 # Choose the larger number
6 if number1 > number2:
7     larger_number = number1
8 else:
9     larger_number = number2
10
11 # Print the result
12 print("The larger number is:", larger_number)
13

```

```

1 # Read two numbers
2 number1 = int(input("Enter the first number: "))
3 number2 = int(input("Enter the second number: "))
4
5 # Choose the larger number
6 if number1 > number2:
7     print("The larger number is:", number1)
8
9 elif number2 > number1:
10    print("The larger number is:", number2)
11
12 #elif :
13 #elif:
14
15 else:
16    print(number1, "is equal to", number2)
17

```

You can add more elif conditions. But only 1 "if" and 1 "else".

#### ✓ Python Program to Check if a Number is Positive, Negative or Zero:

```

1 num = float(input("Enter a number: "))
2
3
4 if num > 0:
5     print("The number is positive.")
6 elif num < 0:
7     print("The number is negative.")
8 else:
9     print("The number is zero.")

```

↗ Enter a number: 2  
The number is positive.

#### ✓ Loops

We are going to come back to loops after we go through 03\_Python Collection Arrays

#### ✓ While Loop

```

1 i = 1
2 while i<6:

```

```
3 print(i)
4 i +=1
```

```
↩ 1
  2
  3
  4
  5
```

Write a code to print all even numbers from 0 to 100.

```
1 my_number = 0
2 while my_number<=100:
3     print(my_number)
4     my_number +=2 #my_number = my_number +2
```

Write a code to print all number from 0 to 100 which divisible by 7. Hint: use if statement

```
1 my_number = 0
2 while my_number<=100:
3     if my_number%7 == 0:
4         print(my_number)
5     my_number +=1
```

1 Start coding or [generate](#) with AI.

## ▼ For Loop

```
1 print("banana", "apple", sep=",")
```

```
↩ banana,apple
```

```
1 fruits = ["apple", "banana", "berry"]
2 for x in fruits:
3     print(x, end=".")
```

```
1 for x in "banana":
2     print(x)
```

```
↩ b
  a
  n
  a
  n
  a
```

```
1 for x in "Yusuf":
2     print(x, end="-")
```

```
↩ Y-u-s-u-f-
```

```
1 for x in "12345":
2     num = int(x)
3     print(num+1)
```

```
↩ 2
  3
  4
  5
  6
```

```
1 print("Numbers from 1 to 50:")
2 for n in range(1, 51):
3     print(n, end=" ")
```

```
↩ Numbers from 1 to 50:
  1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47
```

Write a code to print all even numbers from 0 to 100. (use step argument)

```
1 print("Even numbers from 0 to 100:")
2 for n in range(0, 101, 2):
```

```
3 print(n, end=" ")
```

Write a code to print all number from 0 to 100 which divisible by 7

```
1 print("Multiples of 7 from 0 to 100:")
2 for n in range(0, 101, 7):
3     print(n, end=" ")
```

Write a code to print all odd numbers from 0 to 100. (use step argument)

```
1 print("Odd numbers from 1 to 100:")
2 for n in range(1, 101, 2):
3     print(n, end=" ")
```

```
1 for n in range(0, 101):
2     if n % 3 == 0 and n % 5 == 0:
3         print("FizzBuzz")
4     elif n % 3 == 0:
5         print("Fizz")
6     elif n % 5 == 0:
7         print("Buzz")
8     else:
9         print(n)
```

```
1 #Task to print fizz or buzz or fizzbuzz
2
3 for n in range(1,101):
4     if n%3==0 and n%5==0:
5         print("\n",n,"-FizzBuzz", end="")
6     elif n%3==0:
7         print("\n",n,"-Fizz",end="")
8     elif n%5==0:
9         print("\n",n,"-Bizz",end="")
10    else:
11        print("\n",n,end=" ")
```

```
1 for num in range(1,101):
2     string = ""
3     if num % 3 == 0:
4         string = string + "Fizz"
5     if num % 5 == 0:
6         string = string + "Buzz"
7     if num % 5 != 0 and num % 3 != 0:
8         string = string + str(num)
9     print(string)
10
```

```
1 Start coding or generate with AI.
```

## ✓ Collection Types in Python

### ✓ List

```
1 mylist = ["apple", "banana", "cherry"]
2
3 word = "happy"
4 mylist2 = list(word)
5
6
7 print(mylist)
8 print(mylist2)
```

```
➦ ['apple', 'banana', 'cherry']
   ['h', 'a', 'p', 'p', 'y']
```

```
1 print(type(mylist))
```

```
1 print(type(mylist2))
```



```

1 age = 32
2 mylist3 = list(age)

1 age = [32]
2 print(type(age))

1 customer_info = ["Mr.X", 22, "UK", "12/05/2001", 1.65]
2 print(customer_info)

1 customer_info

```

## Indexing List

```

1 mylist1 = ["a", "b", 10, 20];
2
3 mylist2 = [1, 2, 3, 4, 5, 6, 7 ];
4
5 print("mylist1[1]: ", mylist1[1])
6
7 print ("mylist2[1:5]: ", mylist2[1:5])
8

```

```

↔ mylist1[1]: b
   mylist2[1:5]: [2, 3, 4, 5]

```

how to get [2, 3, 4, 5] using negative indexing

```
1 print(mylist2[-4:-2])
```

```
↔ [4, 5]
```

mylist2 = [1, 2, 3, 4, 5, 6, 7]; -7 -6 -5 -4 -3 -2 -1

```
1 print ("mylist2[1:5]: ", mylist2[-6:-2]) #last index in range -2 will not be printed
```

```
↔ mylist2[1:5]: [2, 3, 4, 5]
```

```

1 #What will be the output?
2 nums = [10, 20, 30, 40, 50, 60, 70, 80, 90]
3 some_nums = nums[2:7]
4 print(some_nums)

```

```
↔ [30, 40, 50, 60, 70]
```

```

1 #Complete the code.
2
3 nums = [10, 20, 30, 40, 50, 60, 70, 80, 90]
4
5 print(nums[0:4])
6
7
8 #Output:
9 #10, 20,30,40
10

```

```
↔ [10, 20, 30, 40]
```

```

1 thislist = ["apple", "banana", "cherry"]
2
3 print(thislist)
4
5 thislist[1] = "blackcurrant"
6
7 print(thislist)
8
9

```

```

↔ ['apple', 'banana', 'cherry']
   ['apple', 'blackcurrant', 'cherry']

```

```

1 thislist[3]="watermelon"
2
3 print(thislist)

```



```
-----
IndexError                                Traceback (most recent call last)
<ipython-input-17-fe4f2cc80e5c> in <cell line: 1>()
----> 1 thislist[3]="watermelon"
      2
      3 print(thislist)

IndexError: list assignment index out of range
```

```
1 thislist.append("orange")
2
3 print(thislist)
```

```
1 print(thislist)
```

```
1 thislist.insert(2, "watermelon")
2
3 print(thislist)
```

```
1 thislist.sort()
```

```
1 print(thislist)
```

```
1 thislist.insert(5, "kiwi")
```

```
1 print(thislist)
```

```
1 thislist.insert(8, "kiwi")
```

```
1 print(thislist)
```

```
1 thislist.insert(7, "kiwi")
```

```
1 thislist.reverse()
```

```
1 print(thislist)
```

```
1 thislist = ["apple", "banana", "cherry"]
2 print(thislist)
3
4 thislist.pop(1)
5
6 print(thislist)
```



```
['apple', 'banana', 'cherry']
['apple', 'cherry']
```

✓ Exercise: Using the created above (thislist) add 5 more items into this list and then remove index from 2:5..

```
1 thislist = ["apple", "banana", "cherry"]
2 print("List before adding")
3 print(thislist)
4
5
6 thislist.append(["orange", "grape", "kiwi", "melon", "pear"])
7
8
9
10 print("List after adding 5 more items:")
11 print(thislist)
12
13
14
15 print("List after removing element at index 1:")
16 print(thislist)
```

```
1 thislist = ["apple", "banana", "cherry"]
2
```

```
3 newlist = ["mango","orange","plum","pineapple", "peach"]
4
5 thislist.extend(newlist)
6
7 print(thislist)
8
9 del thislist[2:5]
10
11 print(thislist)

1 thislist = ["apple", "banana", "cherry"]
2 print("Original list:", thislist)
3
4
5
6 print("After removing index 1:", thislist)
7
8 #thislist.pop(2:5) this will give error. pop method gets only 1 index value
9
10 thislist.append(["orange", "grape", "mango", "pineapple", "kiwi"])
11 print("After adding 5 items:", thislist)
12 thislist = thislist[:2] + thislist[5:]
13 print("After removing items from index 2 to 4:", thislist)
14
15 print(thislist)
```

## ✓ Tuple

```
1 mytuple = ("apple", "banana", "cherry")
2
3 print(mytuple)
4
5 print("1st index in mytuple:", mytuple[1])
6
7 #Write a code to display last item in mytuple.
8
9
```

```
1 #[10:05] Rahana Begum
2
3 mytuple = ("apple","banana","cherry")
4
5 print (mytuple)
6
7 print("1st index in mytuple:", mytuple[2])
8

1 print("Last index in my tuple:", mytuple[-1])

1 print("last index in mytuple:", mytuple[2])
```

```
1 #[10:06] Georgi Zlatev
2
3 print(mytuple[-1])
```

```
1 tuple_1 = (1, 2, 4, 8)
2 tuple_2 = 1., .5, .25, .125
3 print(tuple_1)
4 print(tuple_2)
5
```

```
1 my_tuple = (1, 10, 100, 1000)
2
3 print(my_tuple[0])
4 print(my_tuple[-1])
5 print(my_tuple[1:])
6 print(my_tuple[:-2])
7
8 for elem in my_tuple:
9     print(elem)
10
11 for abc in my_tuple:
12     print(abc)
```

```
1 my_tuple = (1, 10, 100)
2
3 t1 = my_tuple + (1000, 10000)
4
5 t2 = my_tuple * 3
6
7 print(t2)
8 print(len(t2))
9 print(t1)
10
11 print(10 in my_tuple)
12 print(-10 not in my_tuple)
13
```

```
1 print(len("my name is Yusuf"))
```

```
1 my_tuple = (101, 10, 100)
2
3
4 print(my_tuple)
```

## ✓ Set

```
1 mySet= {"apple", "banana", "cherry"}
2
3 print(mySet)
4
```

```
1 print(mySet[1])
```

```
1 mySet2= {"apple", "banana", "cherry", "apple", "banana"}
2
3 print(mySet2)
4
```

## ✓ Dictionary

```
1 mydict = {"brand": "Ford", #brand:key and ford:value
2           "model": "Mustang",
3           "year": 1964}
4 print(mydict)
```

```
1 print(mydict.get("brand"))
```

```
1 print(mydict.get("year")) #access 1964
```

```
1 print(mydict.get(1)) #access 1964
```

```
1 print(mydict.get("Ford"))
```

```
1 dictionary = {"cat": "chat", "dog": "chicken", "horse": "cheval"}
2
3 phone_numbers = {'boss': 5551234567, 'Suzy': 22657854310}
4
5 empty_dictionary = {}
6
7 print(dictionary)
8 print(phone_numbers)
9 print(empty_dictionary)
10
```

```
1 print(dictionary.keys())
```

```
1 print(dictionary.values())
```

