Variables
We use variables to temporarily store data in computer's memory.
x = 10 is an integer (a whole number without a decimal point)
x = 4.9 is a float (a number with a decimal point)
x = 'Python' is a string (a sequence of characters)
x = True is a boolean. Boolean values can be True or False.

Receiving Input
input() -> return a string
number = int(input('enter number :')) -> return number

Strings
We can define strings using single (' ') or double (" ") quotes.
We can get individual characters in a string using square brackets [].
course = 'Python for Beginners'
course[0] # returns the first character -> P
course[1] # returns the second character -> y
course[-1] # returns the first character from the end -> s
course[-2] # returns the second character from the end -> r
We can slice a string using a similar notation:
course[1:5]
The above expression returns all the characters starting from the index position of 1 to 5 (but excluding 5). The result will be 'ytho'
If we leave out the start index, 0 will be assumed.
If we leave out the end index, the length of the string will be assumed.

We can use formatted strings to dynamically insert values into our strings:
name = 'hodi'
message = f'Hi, my name is {name}' -> Hi, my name is hodi

message.upper() # to convert to uppercase
message.lower() # to convert to lowercase
message.title() # to capitalize the first letter of every word
message.find('p') # returns the index of the first occurrence of p
 (or -1 if not found)
message.replace('p', 'q')
contains = 'Python' in course -> True / False

Arithmetic Operations
-
+
*
/ # returns a float
// # returns an int
% # returns the remainder of division
** # exponentiation - x ** y = x to the power of y

a > b
a >= b (greater than or equal to
a < b
a != b (not equals)
a == b (equals)
a <= b

Augmented assignment operator:
x = x + 10
x += 10

If Statements

```
if condition:
    print(“true block”)
elif is_cold:
    print(“else true block”)
else:
    print(“false false block")
```

While loops

```
i = 1
while i < 5:
    print(i)
    i+=1
```

For loops

```
for i in range(1, 5):
    print(i)
```

- range(5): generates 0, 1, 2, 3, 4
- range(1, 5): generates 1, 2, 3, 4
- range(1, 5, 2): generates 1, 3

Lists

```
numbers = [1, 2, 3, 4, 5]
numbers[0] # returns the first item
numbers[1] # returns the second item
numbers[-1] # returns the first item from the end
numbers[-2] # returns the second item from the end
numbers.append(6) # adds 6 to the end
numbers.insert(0, 6) # adds 6 at index position of 0
numbers.remove(6) # removes 6
numbers.pop() # removes the last item
numbers.clear() # removes all the items
numbers.index(8) # returns the index of first occurrence of 8
numbers.sort() # sorts the list
numbers.reverse() # reverses the list
numbers.copy() # returns a copy of the list
```