Python Training Certification Course

Conditional Statements and Functions

simplilearn

Learning Objectives

By the end of this lesson, you will be able to:

- Describe expressions
- Illustrate conditional statements
- Explain loops
- Describe functions

Expressions

Conditional Expressions

Conditional expressions are used for comparison. Conditional statements supported by Python are:

Equal a==b	Not equal a!=b	Less than a<b
Less than or equal a<=b	Greater than a>b	Greater than or equal a>=b

Membership Expressions

Membership expressions are used to validate the membership of a value. It tests for membership in a sequence, such as strings, lists, or tuples. The different membership expressions in Python are:

```
In [14]: a="learning is an art"
         b="art"
         b in a

Out[14]: True

In [15]: b is a

Out[15]: False

In [16]: b is "art"

Out[16]: True
```

Expression:

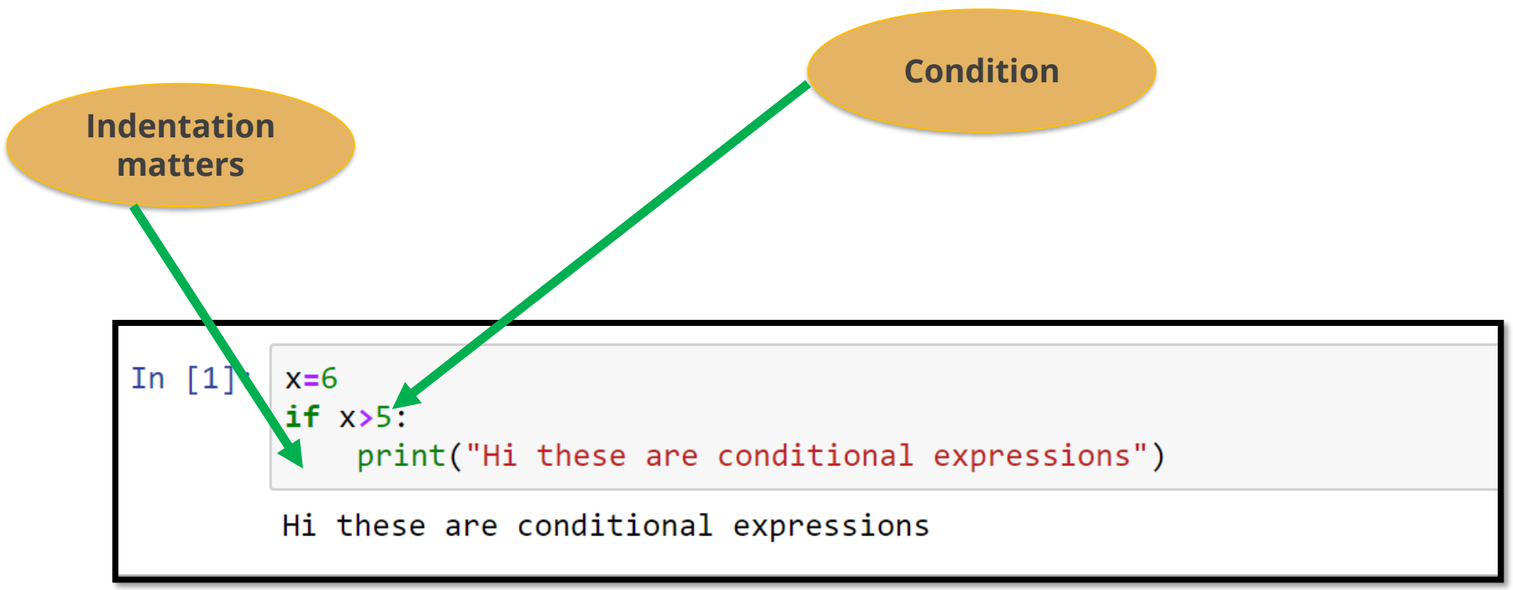
- Checks for equality
- Compares members

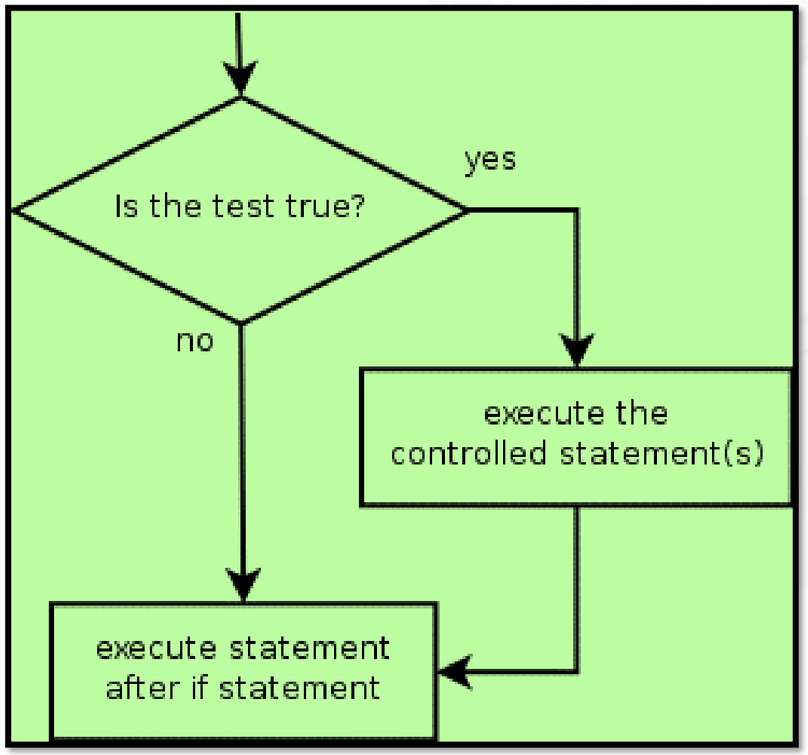
Conditional Statements

if, elif, else

Conditional statements execute a group of statements only based on some condition. In Python, if ,elif, and else are the conditional keyword statements in use.

if: Executes a set of statements only if the condition is true otherwise, statement is skipped.





if, elif, else

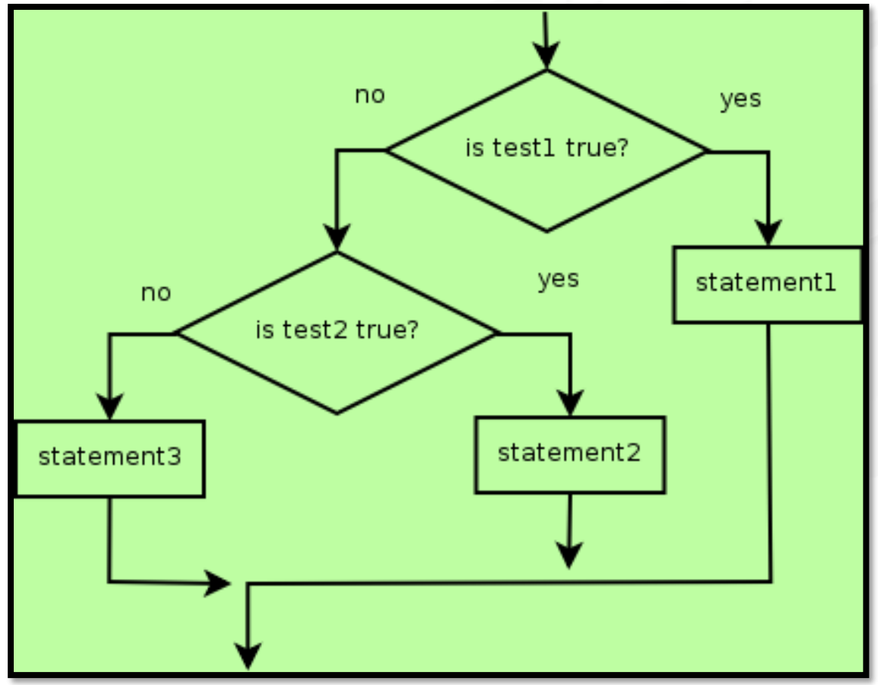
elif: elif or else if is to check multiple conditions.

else: If the condition is not true under if or elif, then the code jumps to else condition.

Jumps to else

```
In [2]: username="pogo123"
        if len(username)>10:
            print("Too lengthly reenter")
        elif len(username)<=4:
            print("Short")
        else:
            print("valid username")

        valid username
```



If else

Duration: 10 min.

Objective: Write a program using Python to demonstrate working of if else.

Steps to demonstrate working of if else:

1. Open Jupyter Notebook
2. Click on File New Notebook
3. Select Python (version 3)
4. Write your program
5. Save your program
6. Click on Run to execute program

ASSISTED PRACTICE

Loops

for Loop

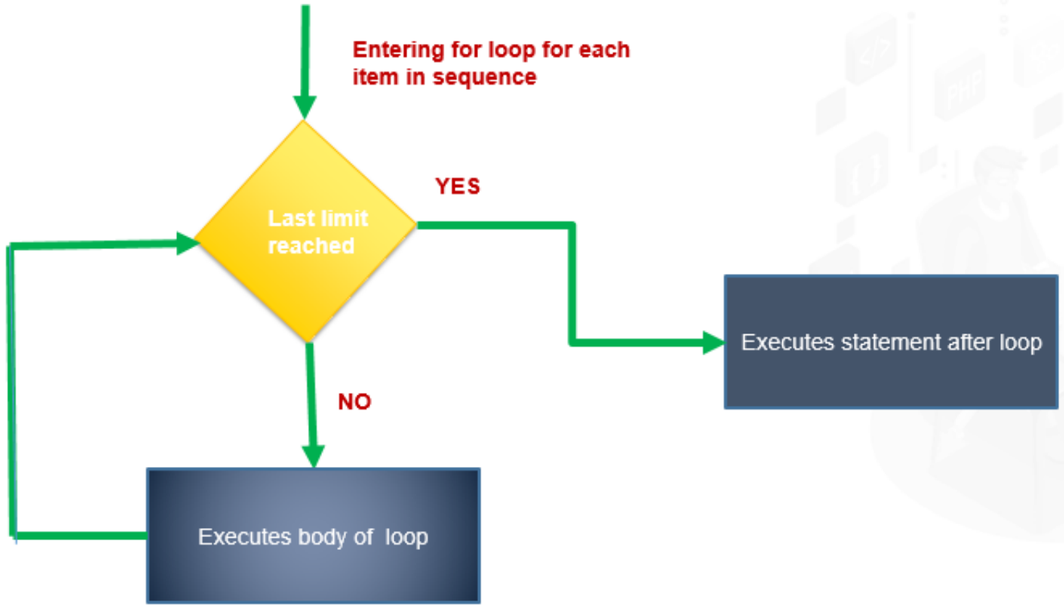
Loops execute a group of statements as long as a condition is **True.** Python has two basic loops: for loop and while loop.

for loop: This is also called iterative loop in Python. For loops can iterate over a sequence of numbers using the "range" and "xrange" functions.

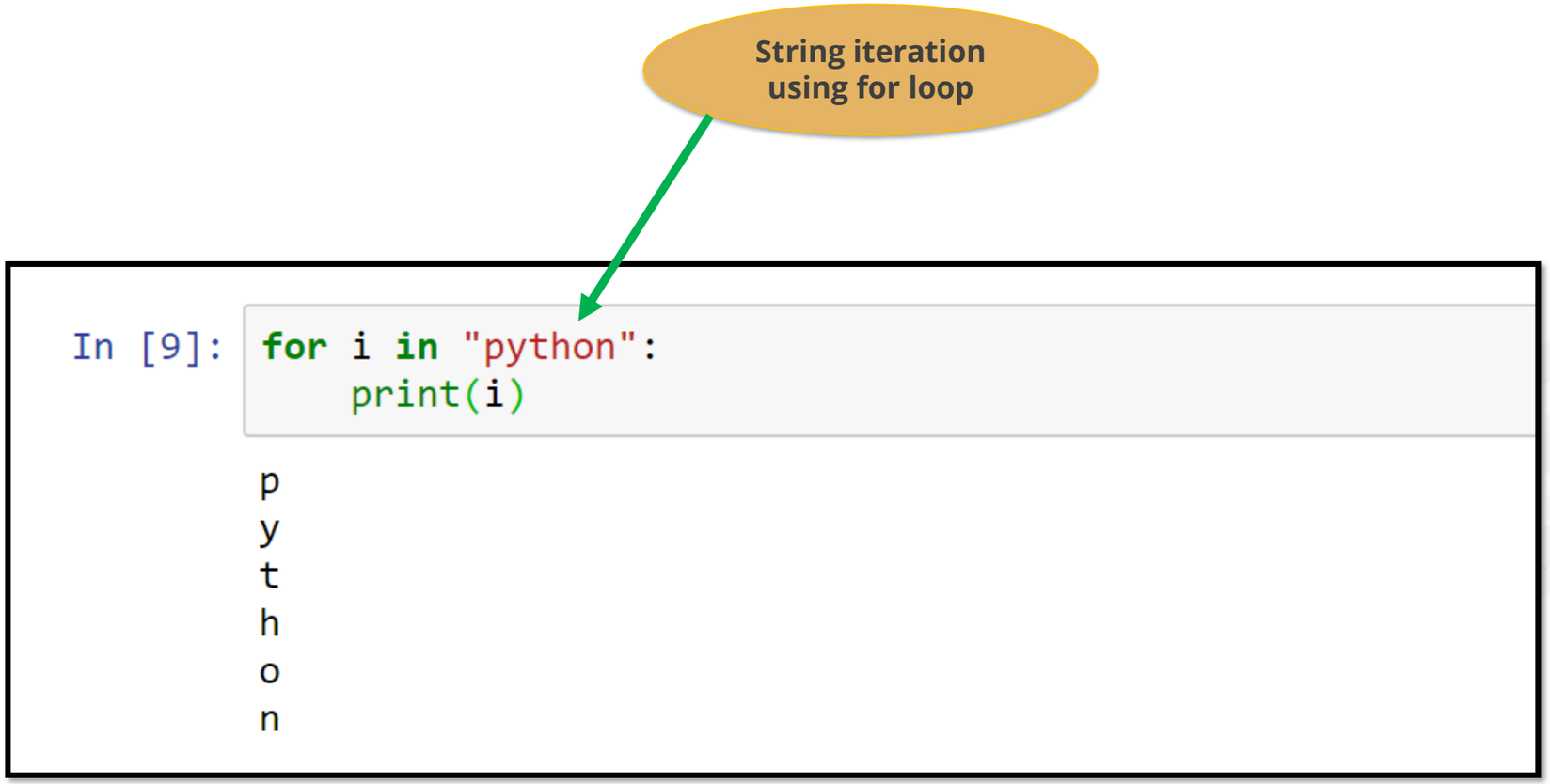
Range function to set sequence limits for iteration

```
In [8]: for x in range(2,11):#start=2 ,end=11
            print(x , end="\t")

2	3	4	5	6	7	8	9	10
```



Sequence Iteration Using for Loop



Sequence Iteration Using for Loop

List iteration using for loop

Tuple iteration using for loop

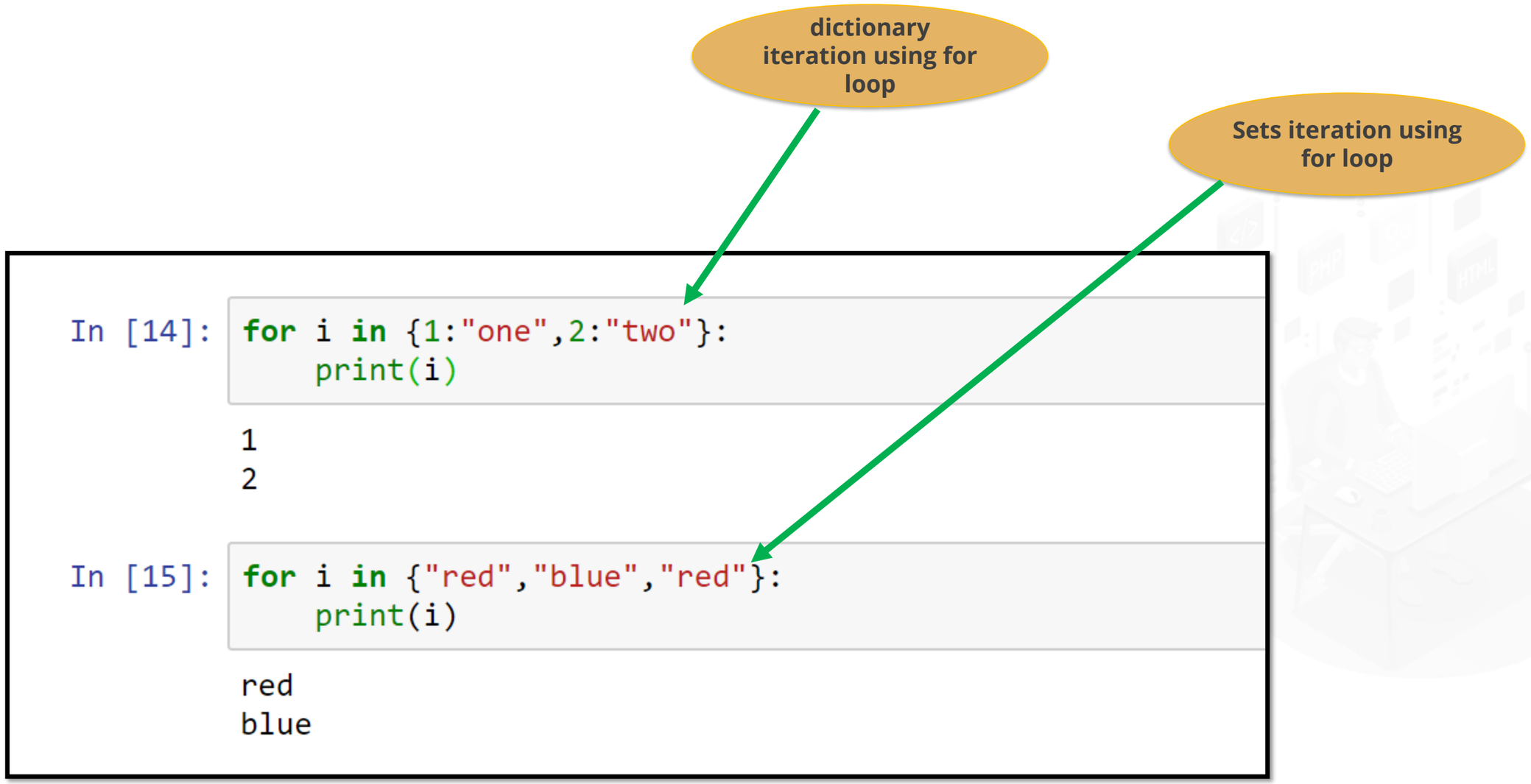
```
In [10]: for i in [1,2,3,4]:
             print(i)

1
2
3
4

In [13]: for i in ("red","blue"):
             print(i)

red
blue
```

Sequence Iteration Using for Loop



For Loop Iterations

Duration: 20 min.

Objective: Write a program using Python to demonstrate working of for loop iterations.

Steps to demonstrate working of for loop iterations:

1. Open Jupyter Notebook
2. Click on File New Notebook
3. Select Python (version 3)
4. Write your program
5. Save your program
6. Click on Run to execute program

ASSISTED PRACTICE

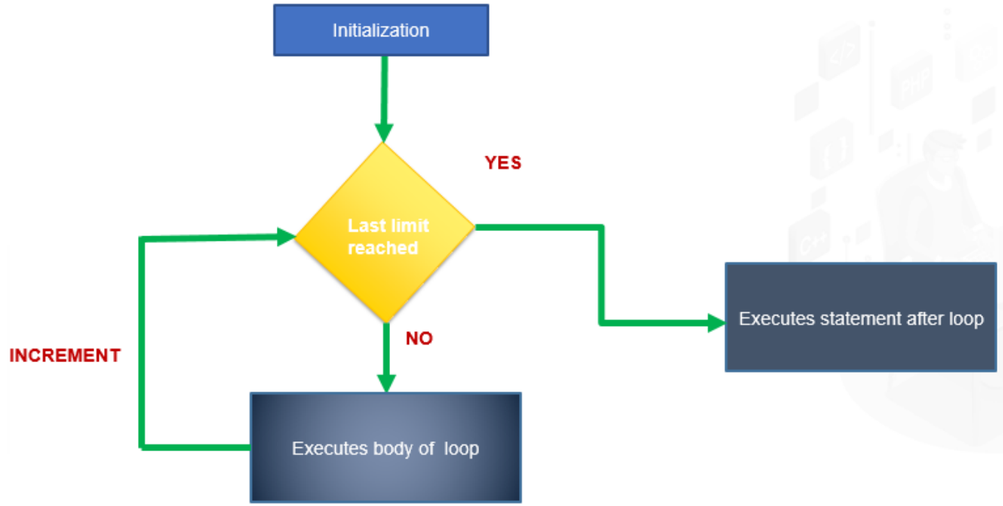
while Loop

while loop: while loops execute a set of statements as long as a condition is true.

```
In [5]: i=0 #initialization
        while i<5:#condition via while loop
            print("Inside loop",i)
            i=i+1
        print("\nend of loop")
```

```
Inside loop 0
Inside loop 1
Inside loop 2
Inside loop 3
Inside loop 4

end of loop
```



while Loop Iterations

Duration: 20 min.

Objective: Write a program using Python to demonstrate working of while loop iterations.

Steps to demonstrate working of while loop iterations:

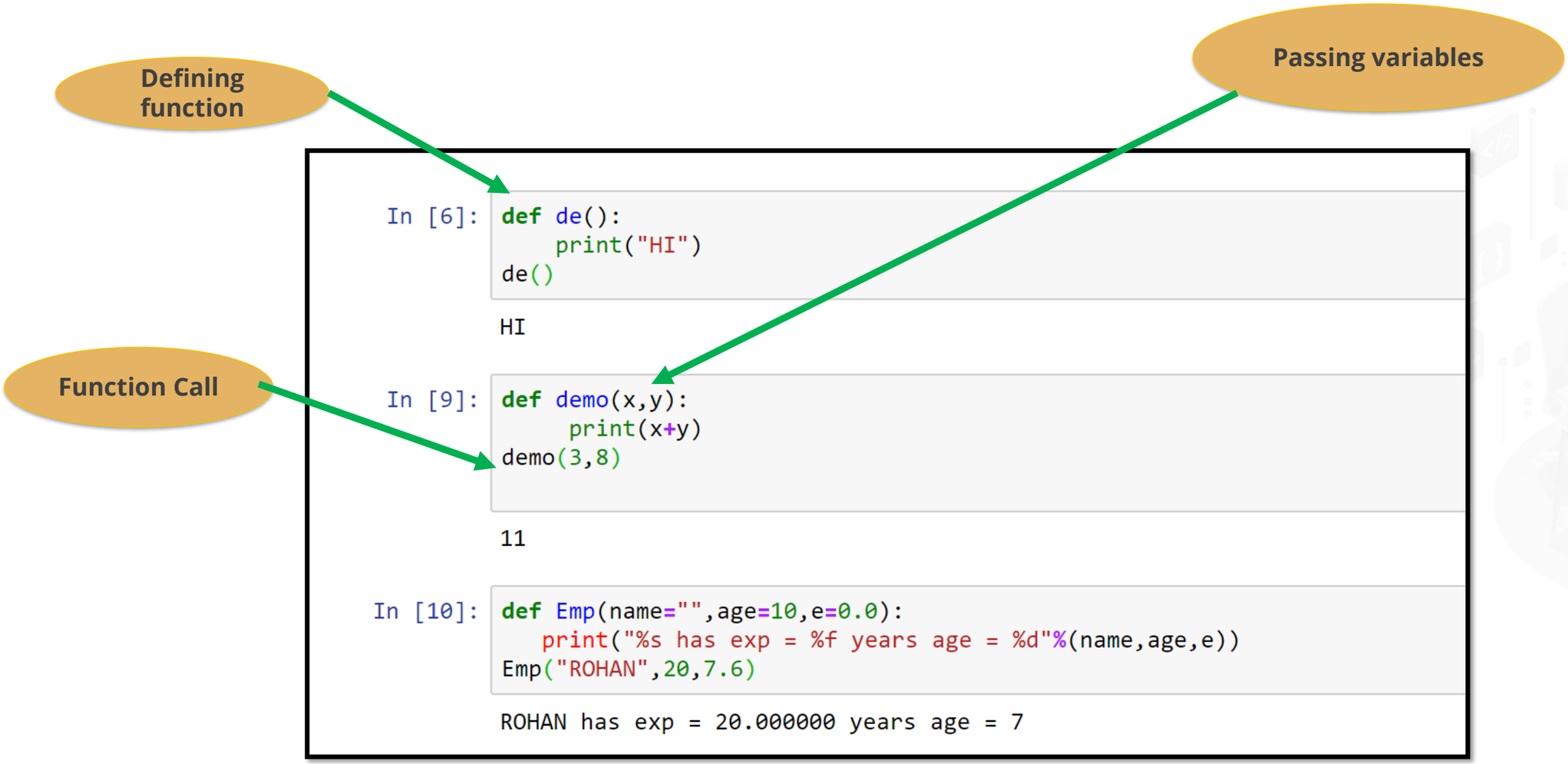
1. Open Jupyter Notebook
2. Click on File New Notebook
3. Select Python (version 3)
4. Write your program
5. Save your program
6. Click on Run to execute program

ASSISTED PRACTICE

Functions

Functions

A function is a block of code which only runs when it is called.



Functions

A return statement ends the execution of the function call and **returns** the result. It is the value of the expression following the return keyword to the caller.

```
In [8]: def add(a,b):
            x=a+b
            return(a+b)
        add(2,7)

Out[8]: 9

In [9]: def add(a,b):
            x=a+b
            return(a)
        add(2,7)

Out[9]: 2
```

In-Built Functions in Python

Python has a large library of in-built functions. Some of them are discussed here.

eval(): It evaluates the passed string as a **Python** expression and returns the result.

lamba(): helps to create one-line functions in Python.

```
In [1]: x=9
        eval("x**3+x**2+3")

Out[1]: 813

In [2]: #Lamda Function : One line functions
        #lambda argument_list: expression
        mul = lambda x, y,z : x * y*z
        mul(3,6,2)

Out[2]: 36
```

In-Built Functions in Python

map() function returns a list of results after applying the given function to each item of a given iterable.

- map(fun, iter) : Syntax
- fun : It is a function to which map passes each element of given iterable.
- iter : It is an iterable which is to be mapped.
- You can pass one or more iterable to the map() function.

In [3]:

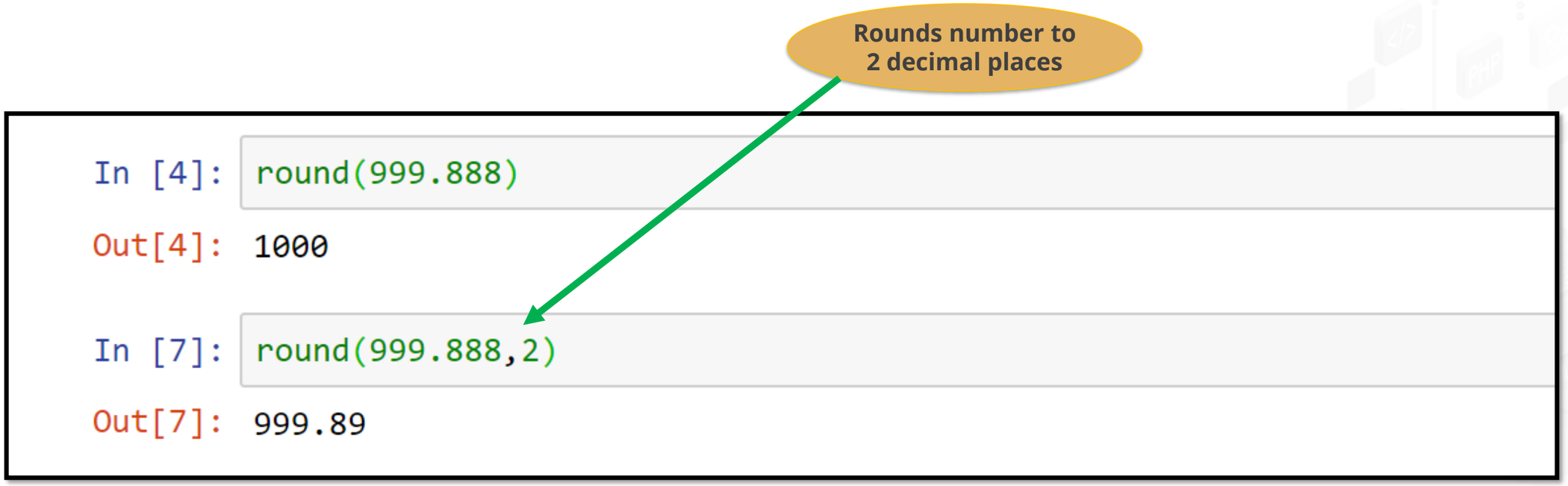
```
# Return square of n
def square(n):
    return n * n

# We double all numbers using map()
numbers = (6, -2, 0, 0.4,5)
result = map(square, numbers)
print(list(result))
```

```
[36, 4, 0, 0.16000000000000003, 25]
```

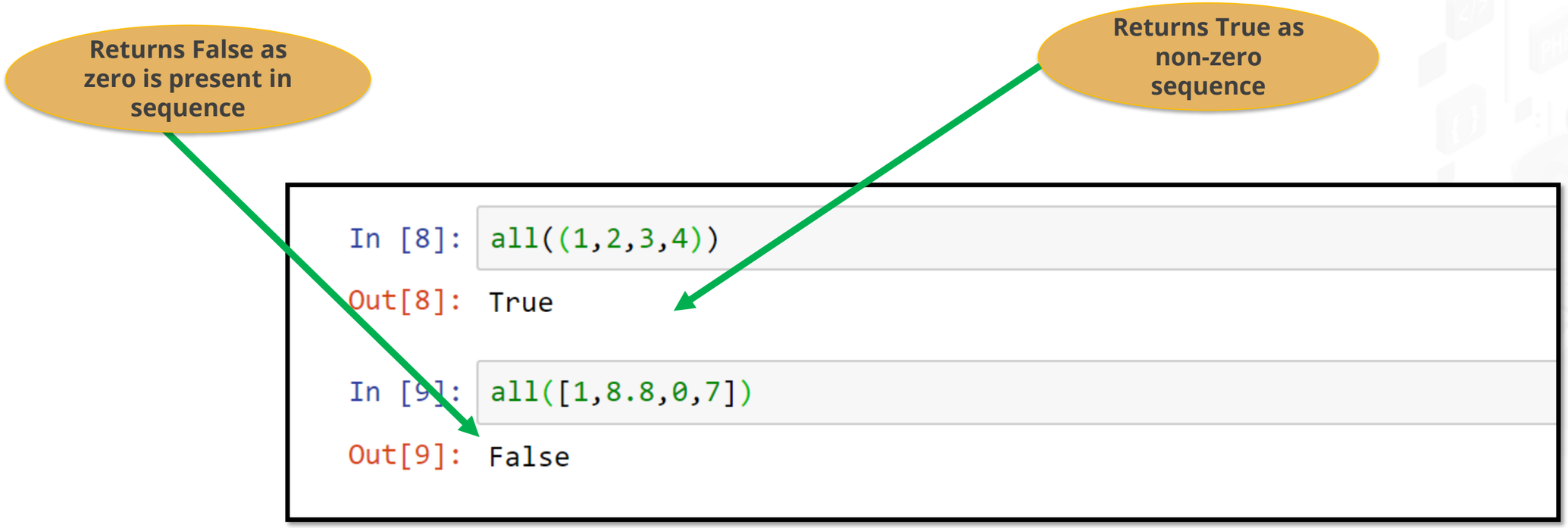
In-Built Functions in Python

round() function rounds a number to give precision in decimal digits. If the number of decimal places has not been specified, the decimal number is rounded off to a whole number.



In-Built Functions in Python

all() function returns false if any one of the elements of the iterable is zero and true if all the elements of the iterable are nonzero.



Functions

Duration: 15 min.

Objective: Write a program using Python to demonstrate working of functions.

Steps to demonstrate working of functions:

1. Open Jupyter Notebook
2. Click on File New Notebook
3. Select Python (version 3)
4. Write your program
5. Save your program
6. Click on Run to execute program

ASSISTED PRACTICE

Knowledge Check

Knowledge Check 1	**“a” is “apple”, will give output as:**

a. a

b. False

c. True

d. Will show error

Knowledge Check 1	**“a” is “apple”, will give output as:**

a. a

b. False

c. True

d. Will show error

The correct answer is **b**

“is “ is a membership operator in Python which returns ‘True’ for exact value.

Knowledge Check 2	Let a="XY" then the expression: for i in a: print(a,end="\t") will give output as:

a. X
Y

b. XY
XY

c. X Y

d. XY XY

Knowledge Check 2	Let a="XY" then the expression: for i in a: print(a,end="\t") will give output as:

a. X
Y

b. XY
XY

c. X Y

d. XY XY

The correct answer is **d**

for loop will iterate the sequence two times and ends each iteration with a tab.

Knowledge Check 3	**eval() function evaluates:**

a. Expression passed as a string

b. Expression only if it contains digits

c. Expression containing digits and strings

d. Expression to sort the strings

Knowledge Check

3

eval() function evaluates:

a. Expression passed as a string

b. Expression only if it contains digits

c. Expression containing digits and strings

d. Expression to sort the strings

The correct answer is **a**

eval() evaluates the expression passed as a string and returns the result.

Knowledge Check 4	The output of: max(False,-4,-1)

a. -1

b. True

c. False

d. -4

Knowledge Check

4

The output of: max(False,-4,-1)

a. -1

b. True

c. False

d. -4

The correct answer is **c**

The function max() is being used to find the maximum value from -4, -1, and false. Since, false amounts to the value zero, Hence, False is the right answer.

Key Takeaways

- Conditional statements use conditional operators.
- For loop is an iterative loop in Python.
- Functions limit the use of loops.