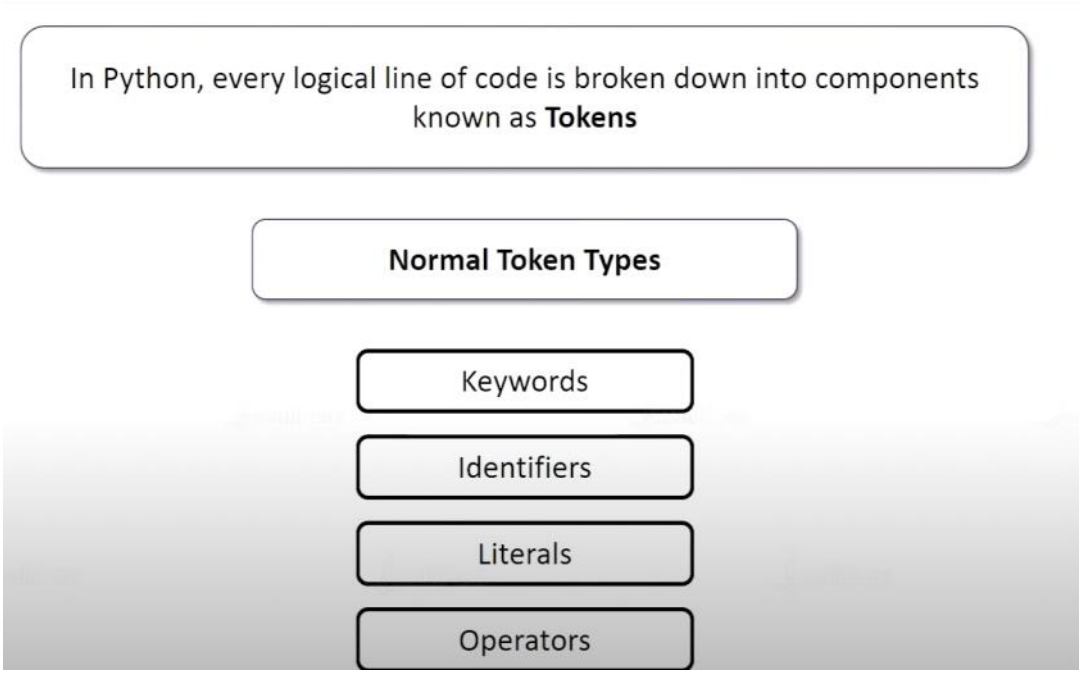
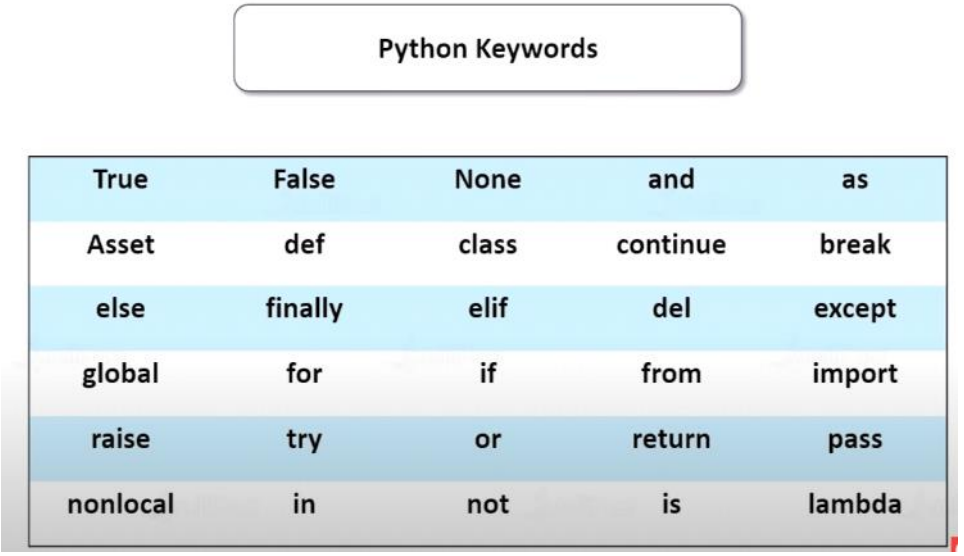
Python tokens

Wednesday, August 17, 2022 2:29 PM

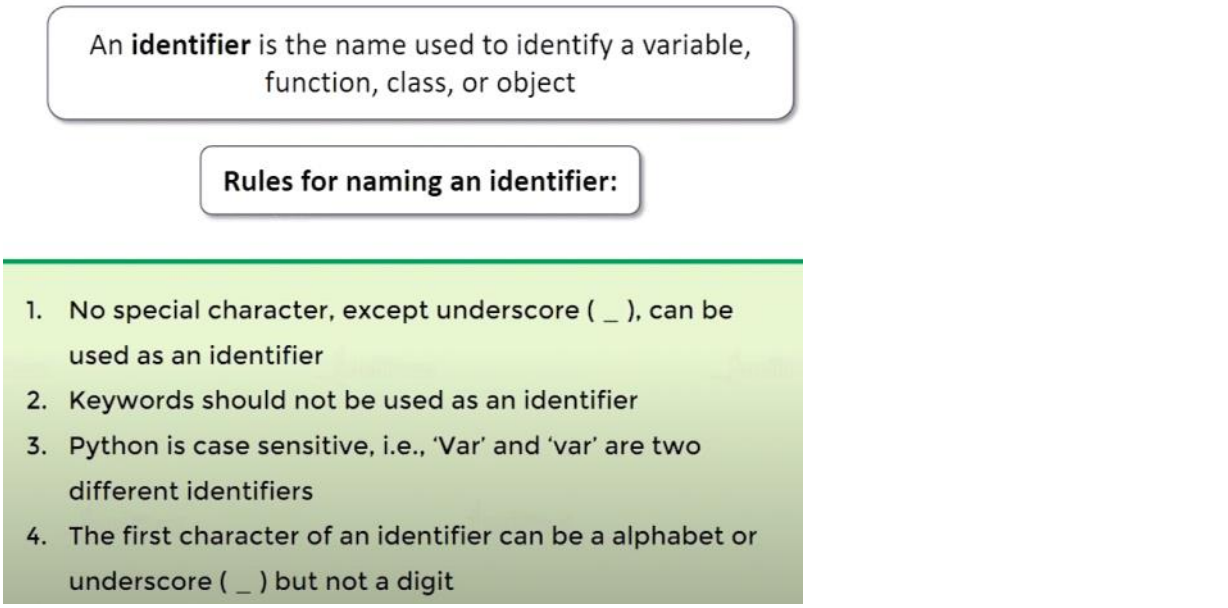
A



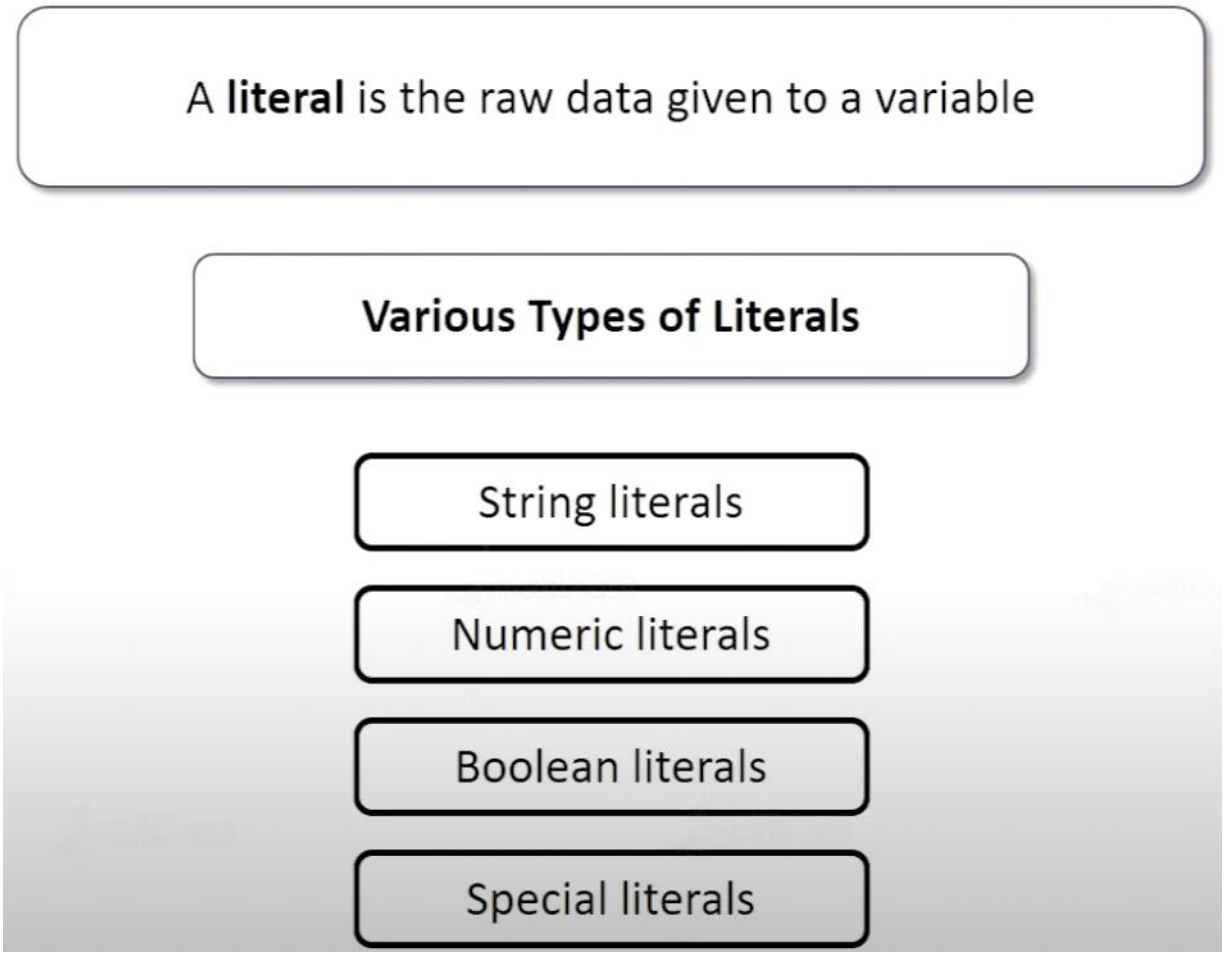
1.Python keywords



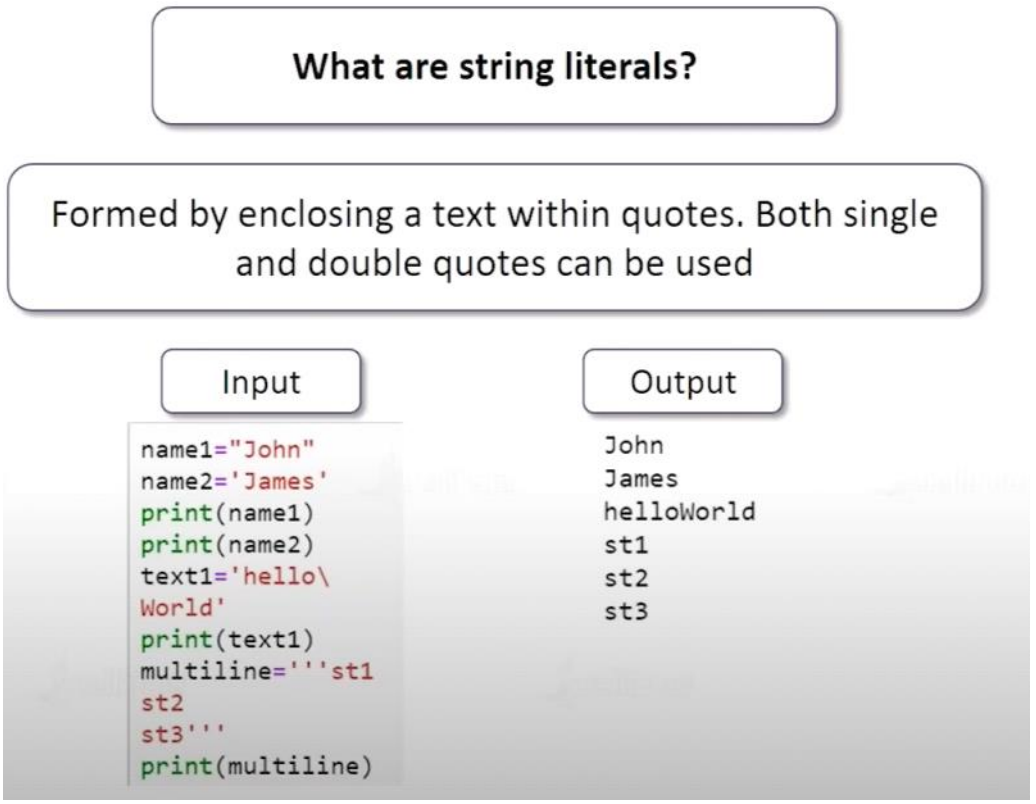
2.Identifier



3.literals



String literals

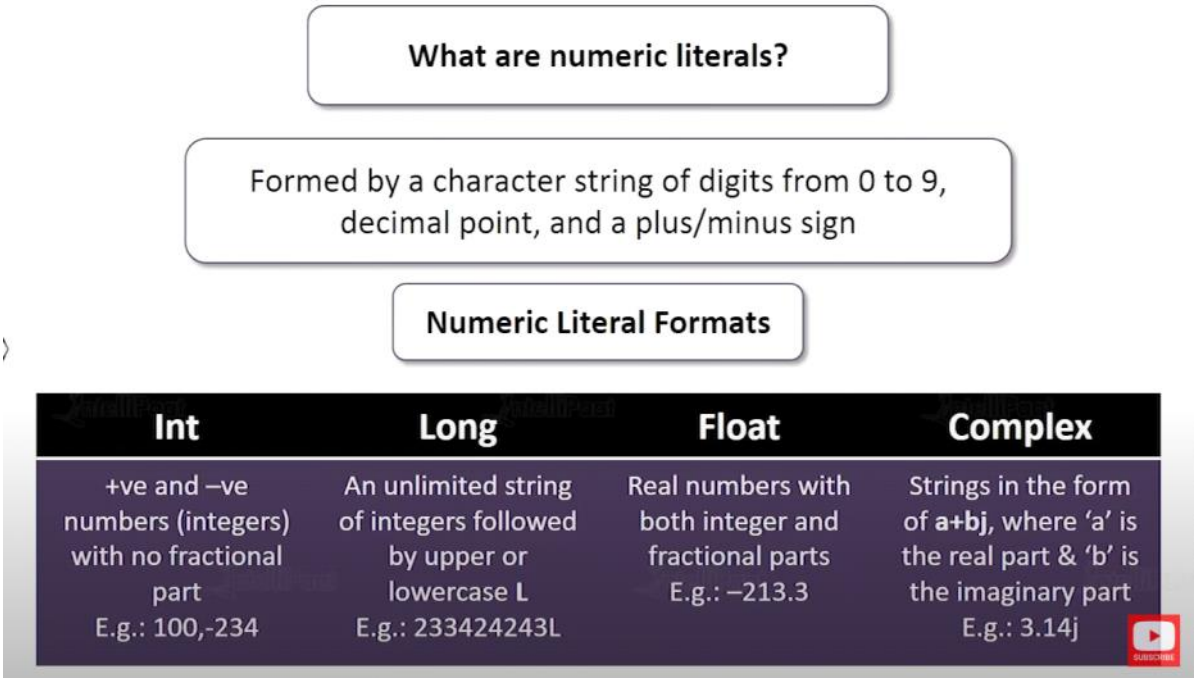


Note

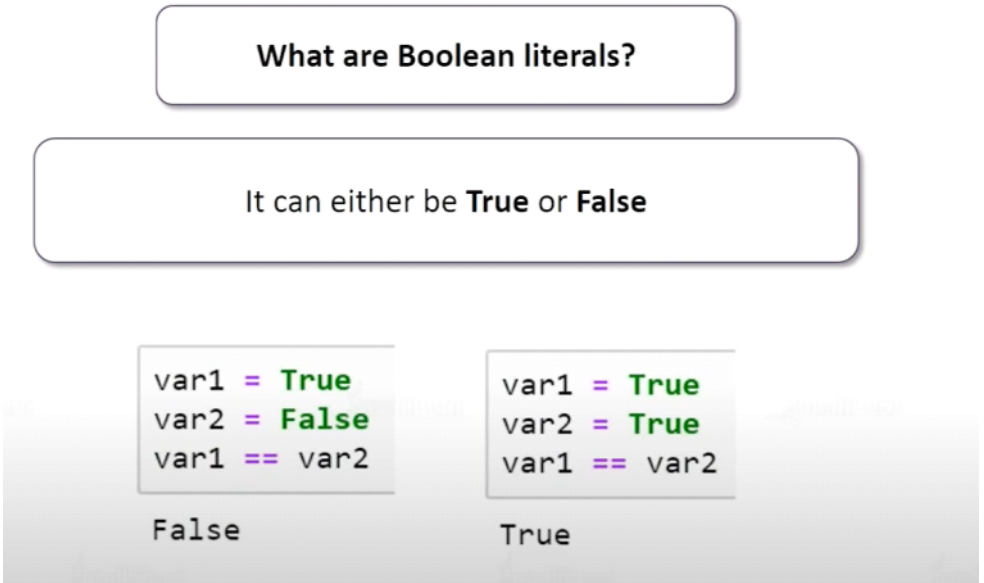
We can also type whole paragraph in the python

```
>>> line = '''
... nnuifwbviwv jsbfuwia cakjssanvcwnavuiwadhc iuwac
... awfuhwacdwaiofjcwaidcidoah dhfiualhfdasio
... cduwhciuwdhiua guafid
... fuhawiduhuiladbiayd ahfuia
... '''
>>> line
'\nnnuifwbviwv jsbfuwia cakjssanvcwnavuiwadhc iuwac\nawfuhwacdwaiofjcwaidcidoah
nfuhawiduhuiladbiayd ahfuia\n'
>>>
```

Numeric literals



Boolean literals



Special literals

What are special literals?

There is a special literal in Python called **None** which means that the variable is yet to be initialized

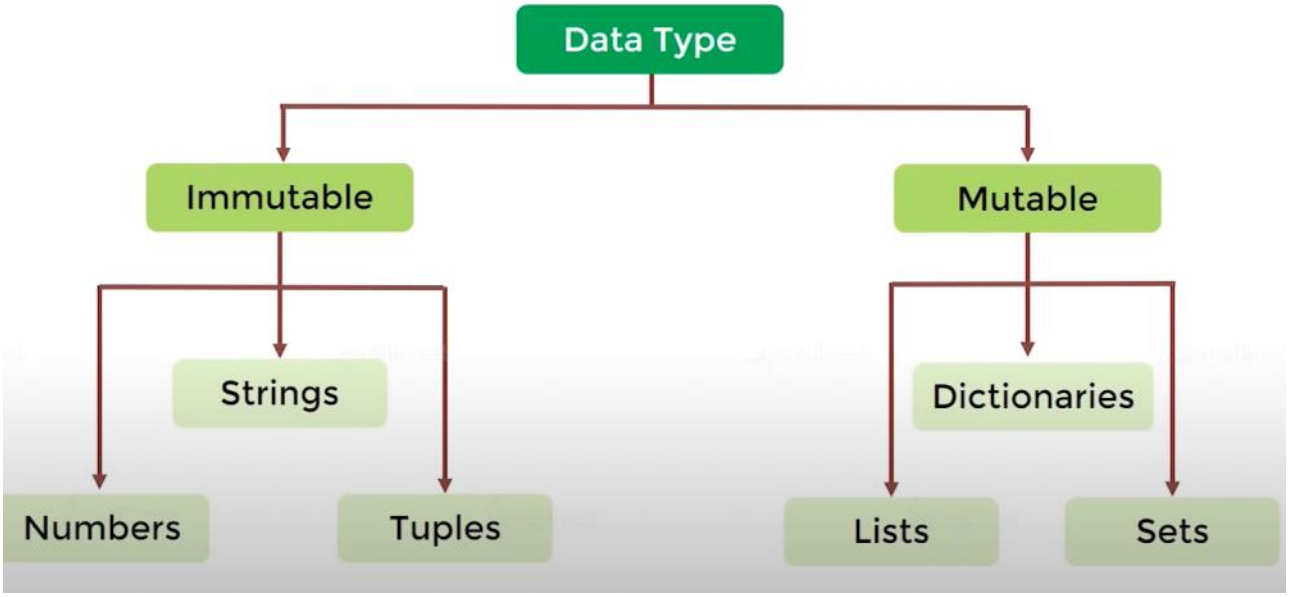
```
val1 = 10
val2 = None
print(val1)
type(val1)
```

```
10

int
```

```
val1 = 10
val2 = None
print(val2)
type(val2)
```

```
None

NoneType
```

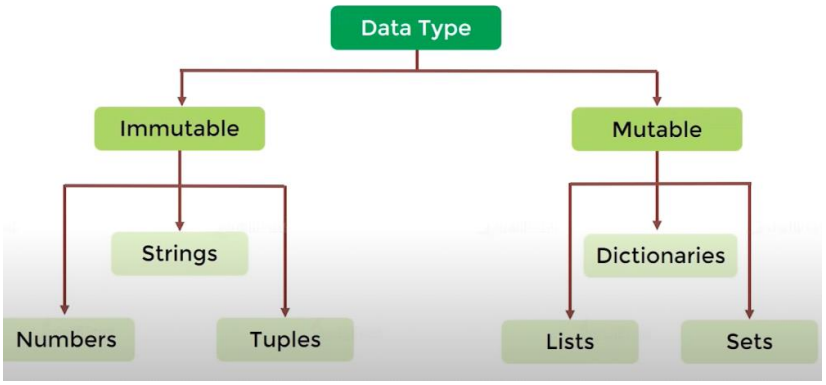
DATA TYPES



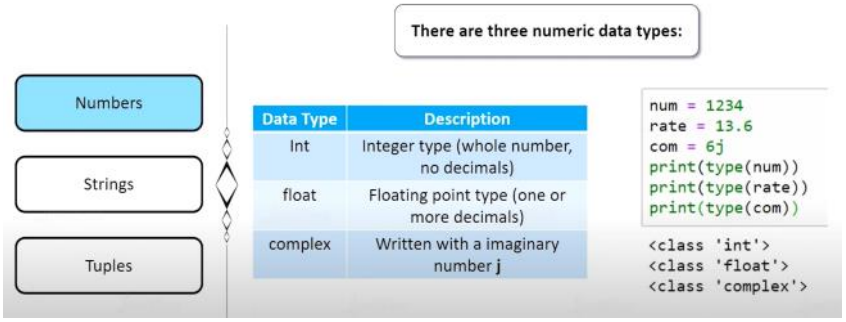
DATA TYPES and also for loop

Thursday, August 18, 2022 10:51 AM

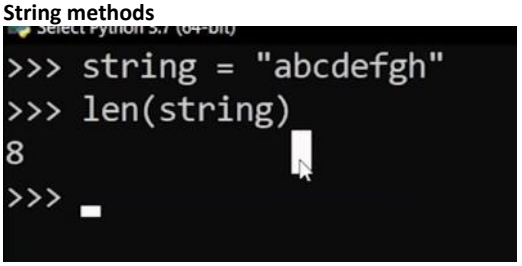
A



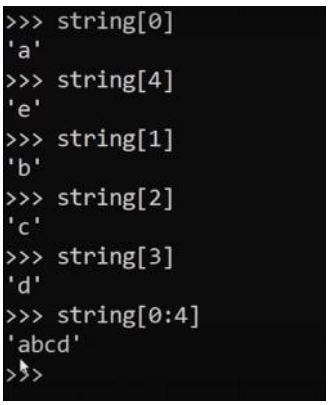
Numbers



String methods



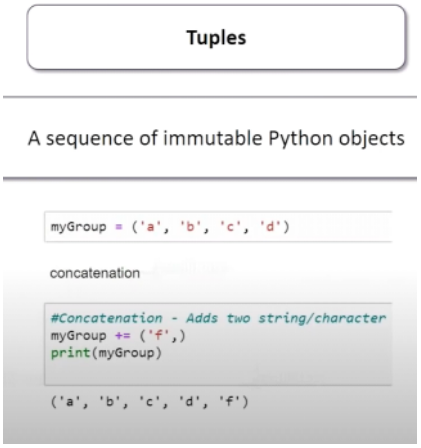
We can find the value by the index number



In the above array we can also perform the replace

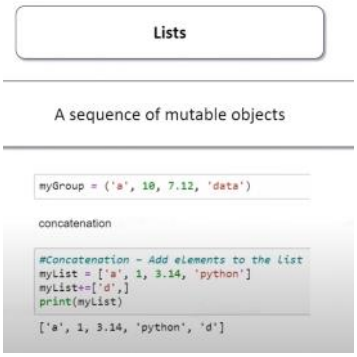
```
>>> string.replace('f', 'g')
'abcdeggh'
>>>
```

Tuples




```
>>> weeks = ("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday")
>>> weeks
('Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday')
```

LIST




```
>>> salaries = [10, 15, 20]
>>> salaries
[10, 15, 20]
>>> type(salaries)
<class 'list'>
```

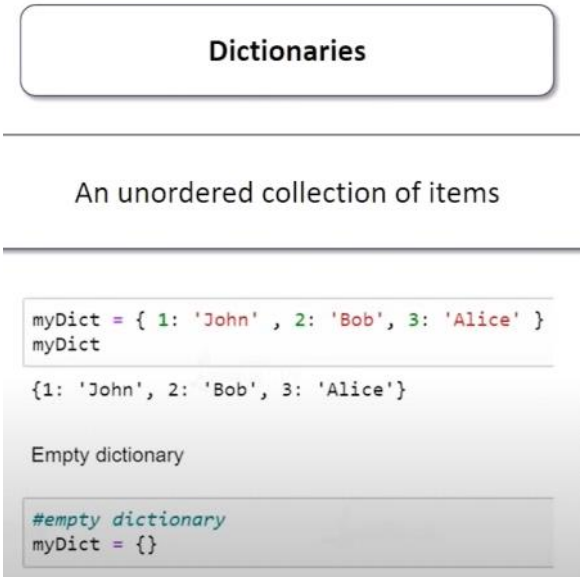
```
>>> salaries
[18, 15, 20]
>>> salaries[0] = 10
>>> salaries[0] = salaries[0] + 8
>>> salaries
[18, 15, 20]
>>> salaries[0] += 8
>>> salaries
[26, 15, 20]
>>> salaries.append(28)
>>> salaries
[26, 15, 20, 28]
```

For tuple we use () and for list we use []

```
>>> mixed = [1, 'a', 1.5, [155, 'b']]
>>> mixed = [1, 'a', 1.5, [155, 'b'], (88, 777)]
>>> mixed
[1, 'a', 1.5, [155, 'b'], (88, 777)]
>>> mixed[4]
(88, 777)
>>> type(mixed[4]
... )
<class 'tuple'>
>>> type(mixed[3])
<class 'list'>
```

The above image justify that in a list we can save the tuple as you can see we saved a int string and float with the list in it and a tuple in it

Dictionaries



```
>>> salaries = { "John": 15, "Jane": 14, "Johnny": 5 }
>>> salaries
{'John': 15, 'Jane': 14, 'Johnny': 5}
>>> type(salaries)
<class 'dict'>
>>> salaries['Jane']
14
>>> salaries['John']
15
```

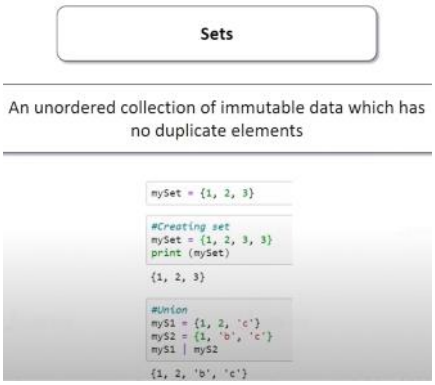
```
>>> salaries.get("John", 15)
15
```

Here we are using get method where I want value of john if john is there in dictionary it will return john value or if there is no john it will give 15

```
>>> salaries.get("Johnasd", 18)
18
```

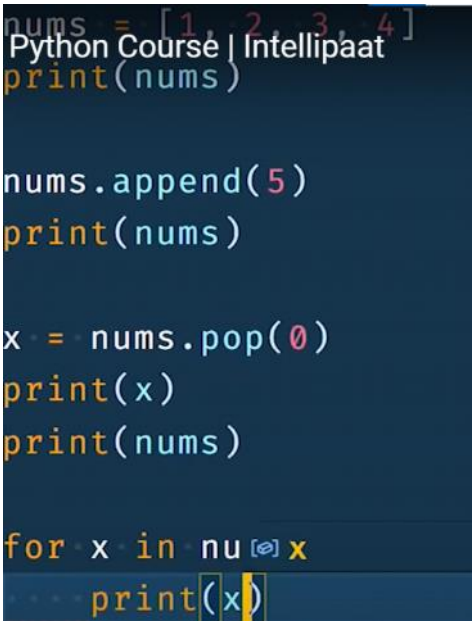
As you can see we do not have Johnasd so we get 18

Sets



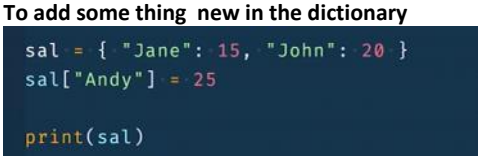

```
>>> s = set()
>>> type(s)
<class 'set'>
>>> s.add(1)
>>> s
{1}
>>> s.add(2)
>>> s
{1, 2}
>>> s.add(2)
>>> s
{1, 2}
>>> 1 in s
True
```

For loop in python

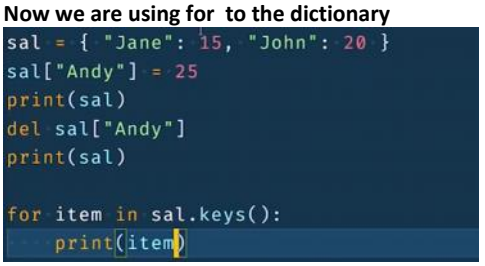


Then it will read the values in the array

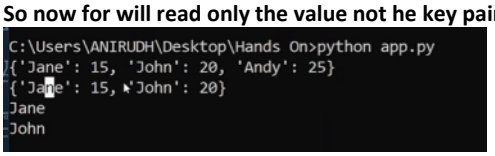
To add some thing new in the dictionary



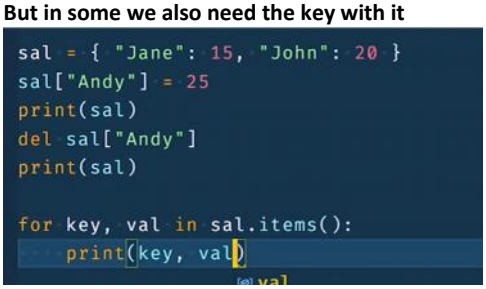
Now we are using for to the dictionary



So now for will read only the value not he key pair of it like it will read john and jane



But in some we also need the key with it



```
C:\Users\ANIRUDH\Desktop\Hands On>python app.py
{'Jane': 15, 'John': 20, 'Andy': 25}
{'Jane': 15, 'John': 20}
Jane 15
John 20
```

Here I am writing a program which will save automatically numbers in the empty list

```
l = []

for x in range(0, 10):
    l.append(x)

print(x)
```

We need to print l not x

```
C:\Users\ANIRUDH\Desktop\Hands On>python app.py
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

Compressed version of above code

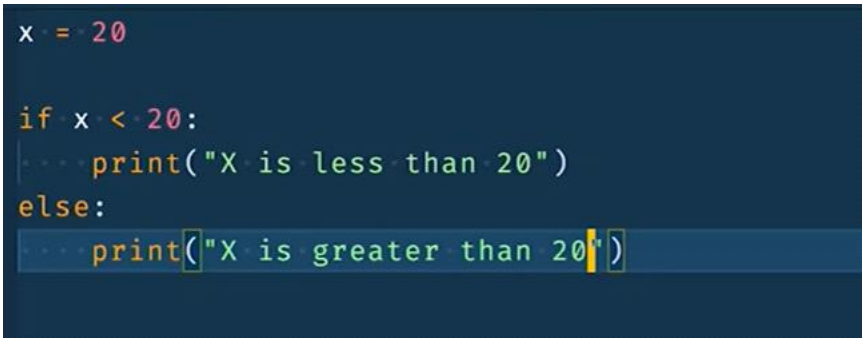
```
l = [x for x in range(0, 9)]

print(l)
```

Conditional statement if else

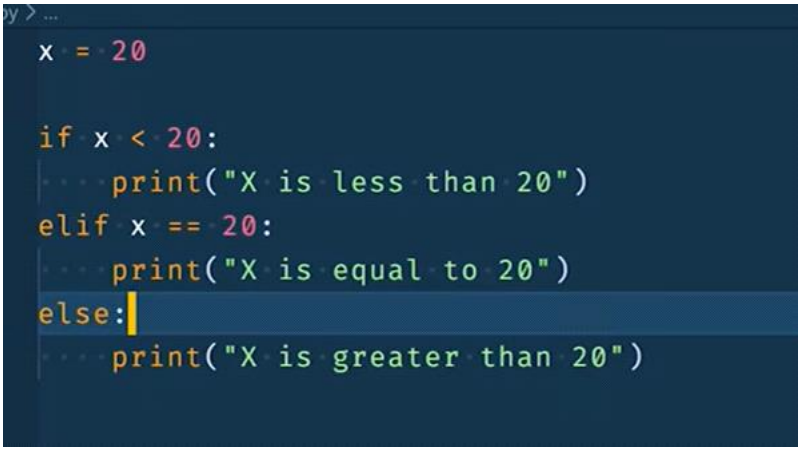
Thursday, August 18, 2022 12:03 PM

A

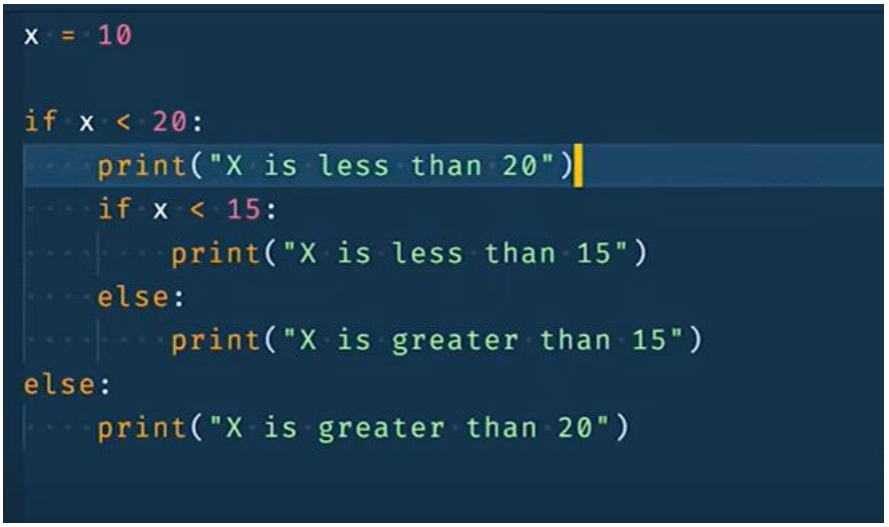
If else



Elif

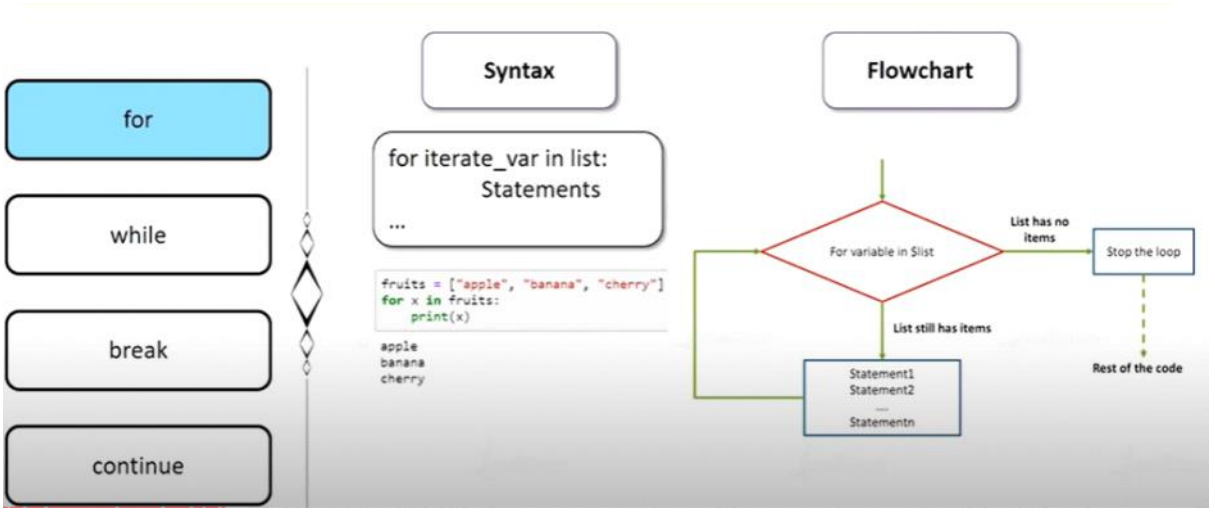


Nested if else


x = 10
if x < 20:
print("X is less than 20")
if x < 15:
print("X is less than 15")
else:
print("X is greater than 15")
else:
print("X is greater than 20")

Loops

Thursday, August 18, 2022 12:12 PM



Using for and break

```
nums = [x for x in range(0, 10)]

for x in nums:
    if (x == 4): break
    print(x)

print("Done")
```

Output will be 0 1 2 3 4

For with continue

```
nums = [x for x in range(0, 10)]

for x in nums:
    if (x == 4): continue
    print(x)

print("Done")
```

here it will skip the 4
Output 0 1 2 3 5 6 7 8 9 10

While

```
a = 10

while a != 0:
    print(a)
    a -= 1
```

Here value is 10
Then we need to read it up to 1
And then a-=1 so slowing value will decrease by 1

So output will be 10 9 8 7 6 5 4 3 2 1

Functions in Python

Friday, August 19, 2022 10:13 AM

A

There are totally 2 function

1. **User define function**
2. **Built in function**

User define function

Example

```
def add (a,b):
    sum = a + b
    return sum
```

these are commonly used in the code

Build in functions

A function already available in a language that we can directly use in our code

abs(): Returns the absolute value of a number

all(): Returns True if all items in an iterable object are true

any(): Returns True if any item in an iterable object is true

ascii(): Returns a readable version of an object and replaces non-ASCII characters with an 'escape' character

bin(): Returns the binary version of a number

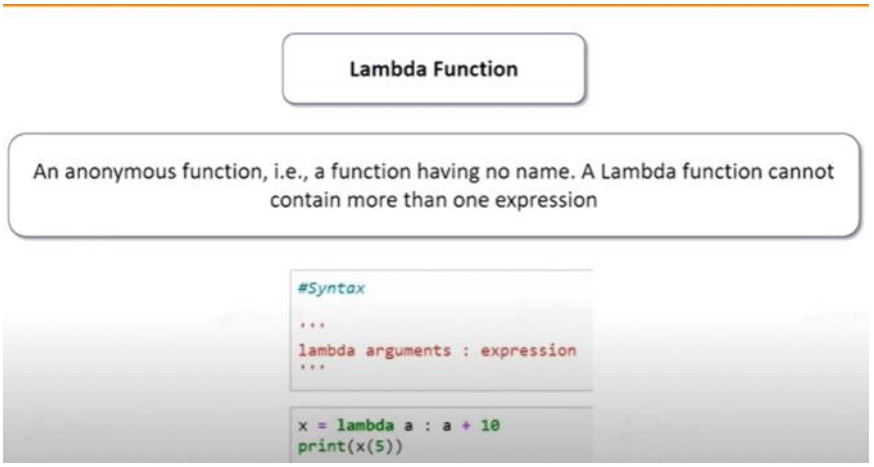
bool(): Returns the Boolean value of a specified object

```
C:\Users\akshay.kanemoni>python
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May  3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> abs(10.234)
10.234
>>> any("string")
True
>>> any(10)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'int' object is not iterable
>>> ascii("A")
"'A'"
>>> ascii(1)
'1'
>>> bin(1)
'0b1'
>>> bin("a")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'str' object cannot be interpreted as an integer
>>> bool("string")
True
>>>
```

The main difference between the ANY and the All is in all the series which we give everything should be true

In any there is no need the series things should all be true we can also give some false one

Lambda functions



There is also a facility where we can use a function in a lambda function

```
lambda.py > ...
1 def myfunc(n):
2     return lambda a: a+n
3 mySum = myfunc(3)
4 print(mySum(10))

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  JUPYTER

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/psco

PS D:\akshay.kanemoni\OneDrive - Infosys Limited\Document
9/python.exe" "d:/akshay.kanemoni/OneDrive - Infosys Limi
13
PS D:\akshay.kanemoni\OneDrive - Infosys Limited\Document
```

An another way to describe the lambda functions

```
6 r = lambda x,y:x*y
7 print(r(12,3))
8

PROBLEMS  OUTPUT  DEBUG CONSOLE

Windows PowerShell
Copyright (C) Microsoft Corporatio

Try the new cross-platform PowerSh

PS D:\akshay.kanemoni\OneDrive - I
9/python.exe" "d:/akshay.kanemoni/
36
```

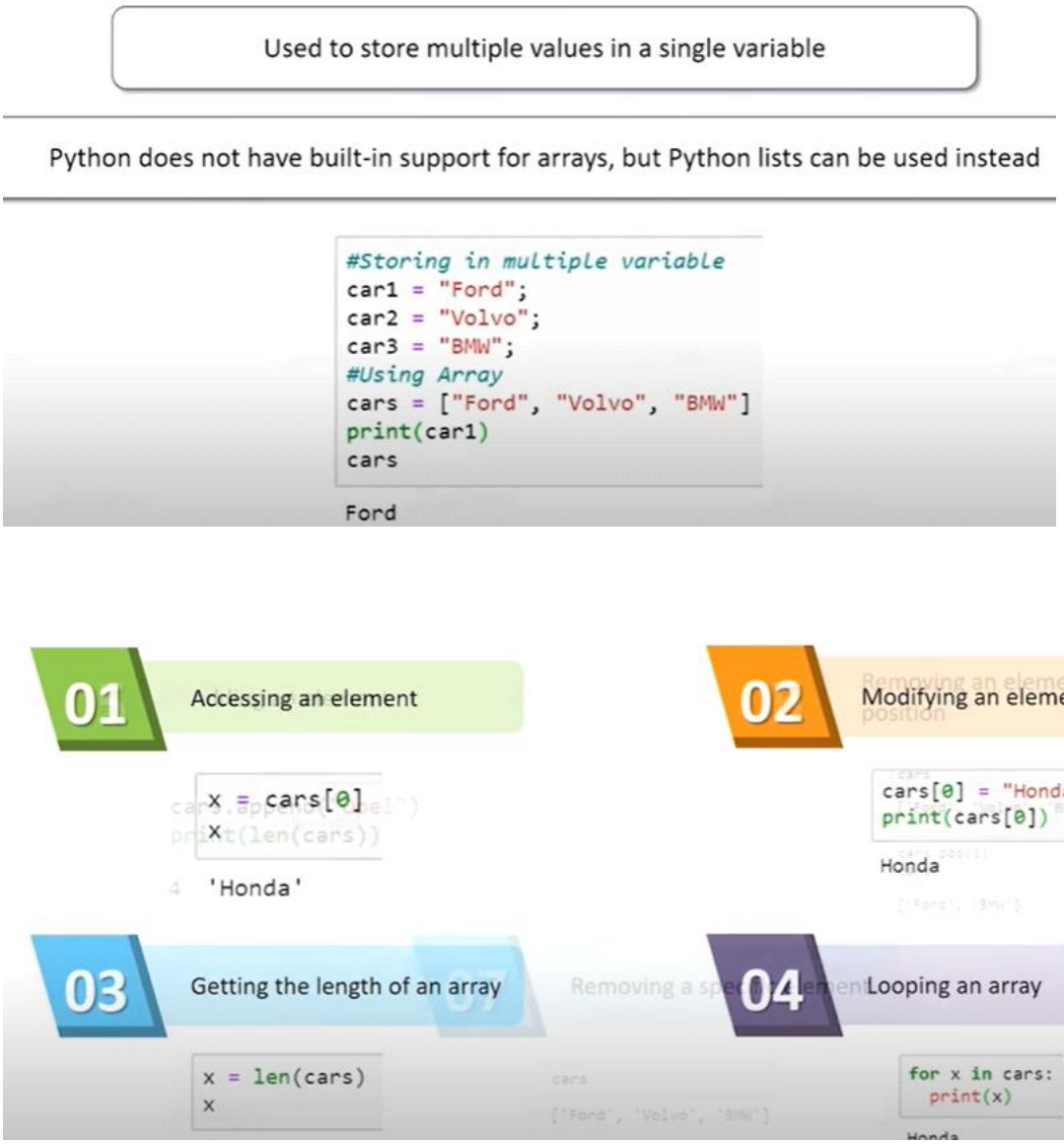
```
def createMultiplier(x):
    return lambda y: x * y

multiply = createMultiplier(10)

def execute(f, arg):
    print("Called F with " + str(arg))
    return f(arg)

print(execute(multiply, 15))
print(execute(multiply, 25))
```

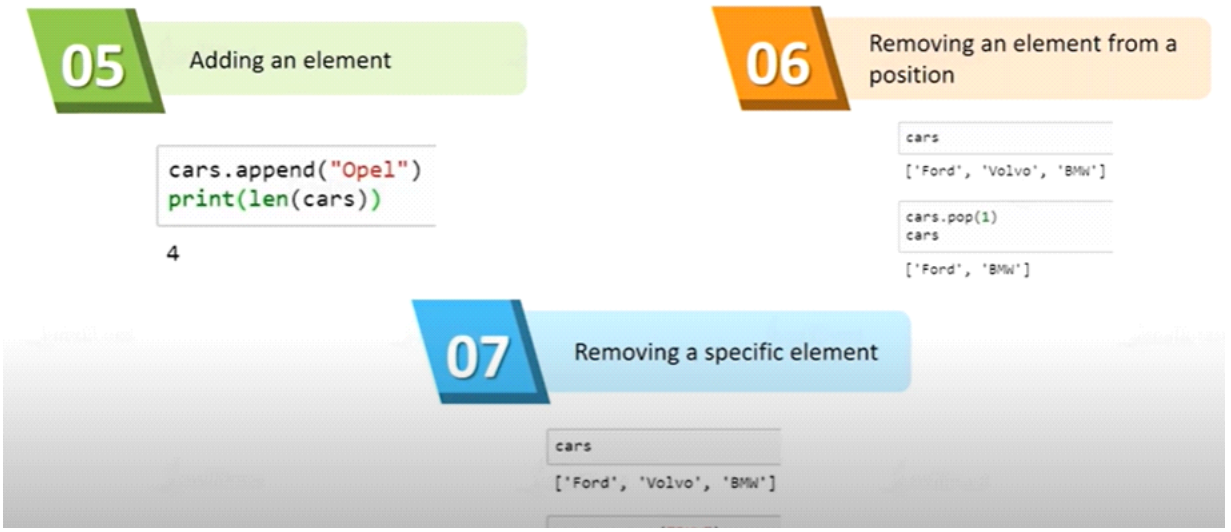
Here f = multiply then f = args = multiply= y

Arrays in python

Friday, August 19, 2022 11:06 AM

Arrays in python




05
Adding an element
cars.append("Opel")
print(len(cars))
4
06
Removing an element from a position
cars
['Ford', 'Volvo', 'BMW']
cars.pop(1)
cars
['Ford', 'BMW']
07
Removing a specific element
cars
['Ford', 'Volvo', 'BMW']

Files in Python

Friday, August 19, 2022 11:12 AM

A

What is a class and what is an object in Python?

1. Python is an object-oriented programming language
2. Almost everything in Python is an object, with its properties and methods
3. A class is like a 'blueprint' for creating objects

Class

```
class MyClass:
  x = 5
```

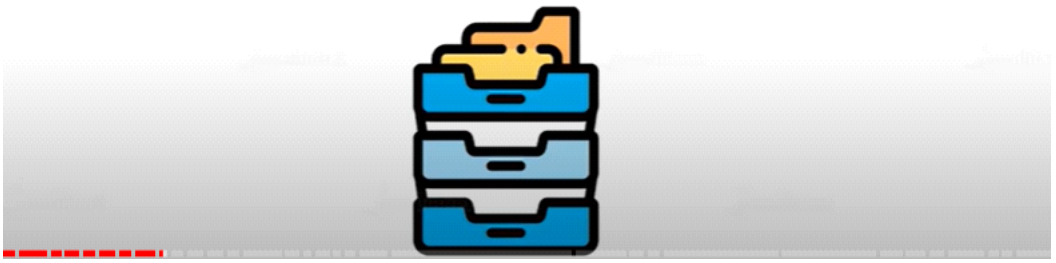
Object

```
obj1 = MyClass()
print(obj1.x)
```

```
5
```

Why do we need file handling?

File handling is important in any application that handles permanent data. We will need file handling if we have to read from or write to files



There are 4 aspects in the file management
1.read
2.Write/Create
3.Open
4.Delete

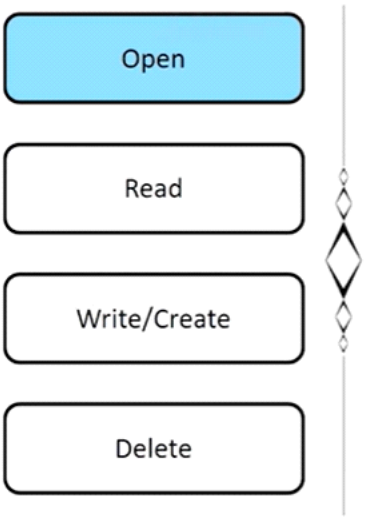
Create

To create a new file

- **'x' – Create**: Creates a file; returns an error if the file already exists

```
# Create a new file
f = open("myfile.txt", "x")
```

Open



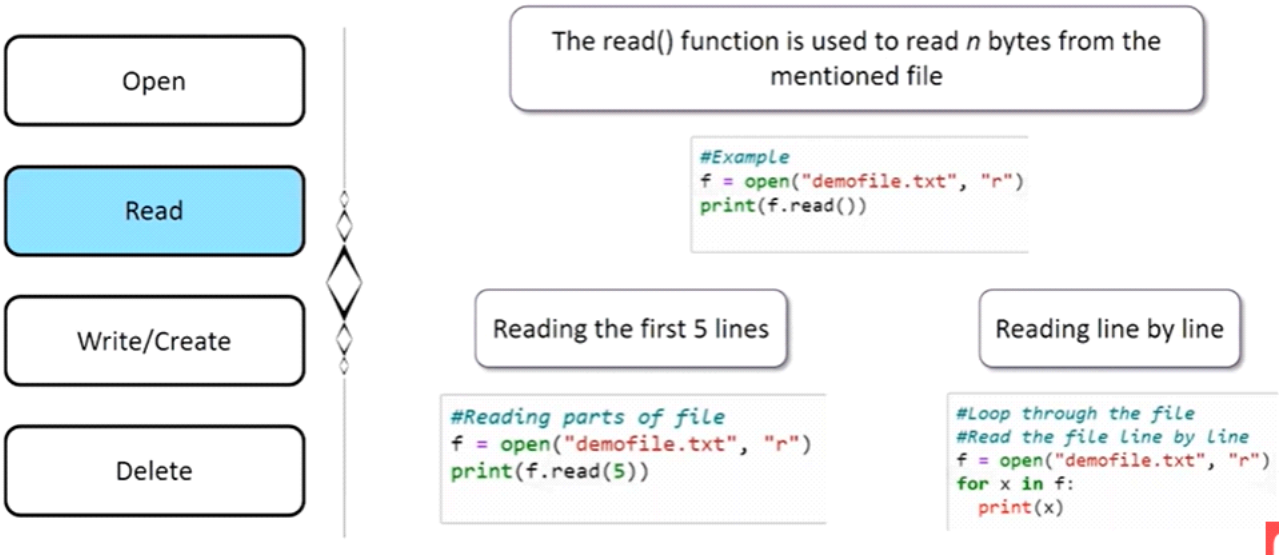
The open() function takes two parameters: filename and mode

```
#Syntax
f = open("path of file")
```

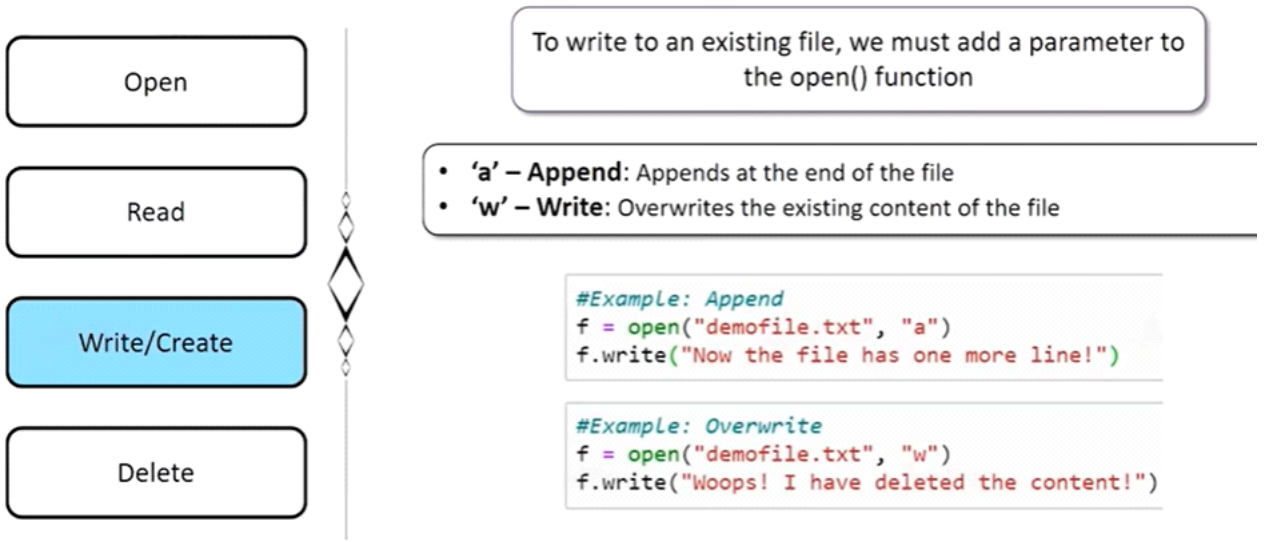
Mode Options

- **'r' – Read**: The default value; opens a file for reading; returns an error if the file does not exist
- **'a' – Append**: Opens a file for appending; creates the file if it does not exist
- **'w' – Write**: Opens a file for writing; creates the file if it does not exist
- **'x' – Create**: Creates the specified file; returns an error if a file with the same name already exists

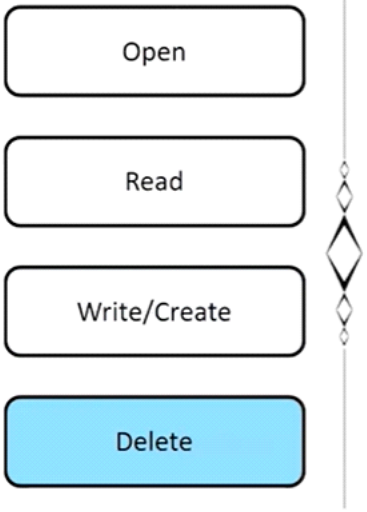
Read



Write



Delete



To import the OS module

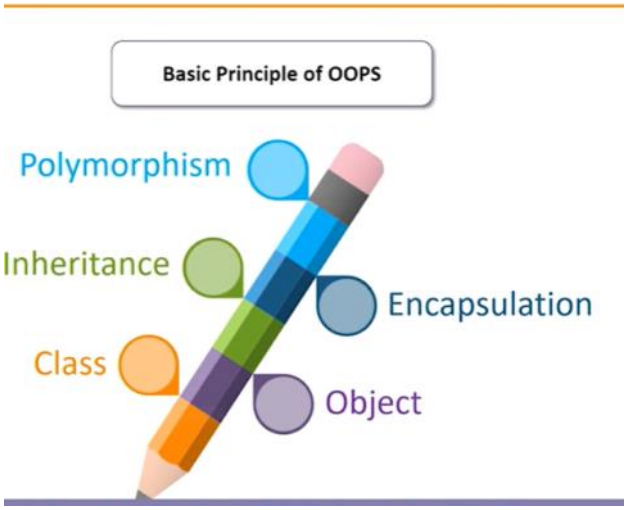
Use the remove() function to delete the mentioned file

```
# Deleting the file
import os
os.remove("demofile.txt")
```

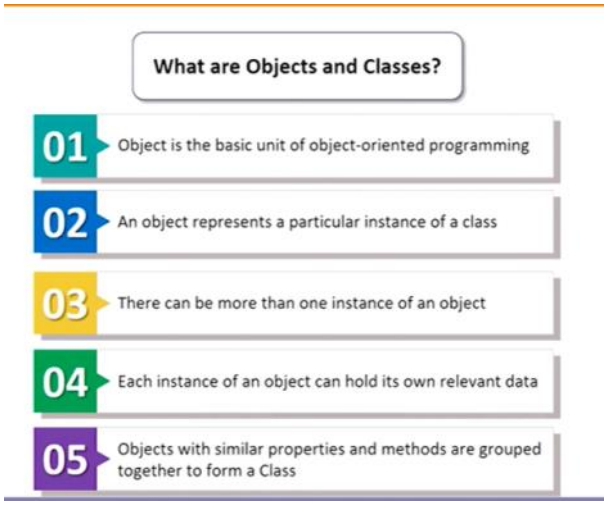
OOPS In python

Friday, August 19, 2022 11:36 AM

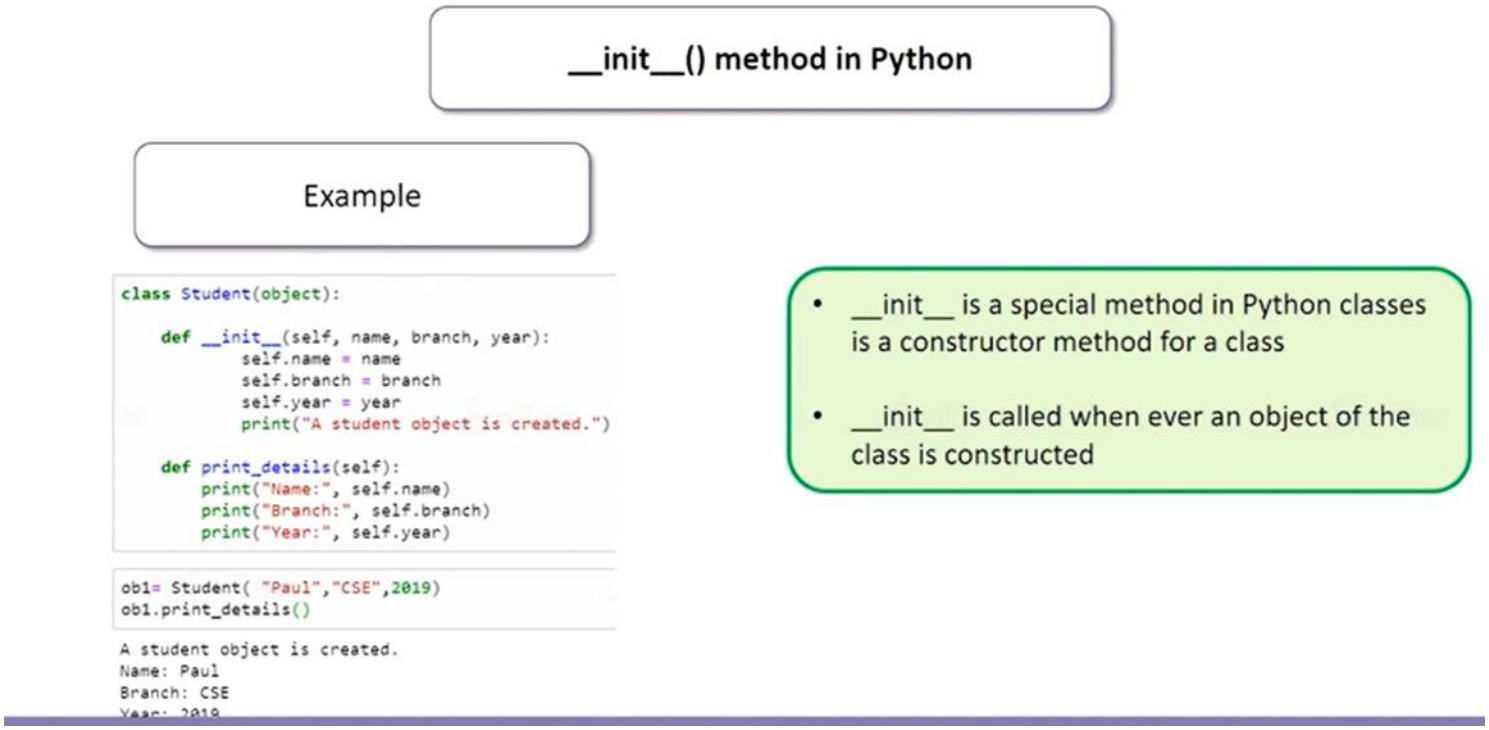
A

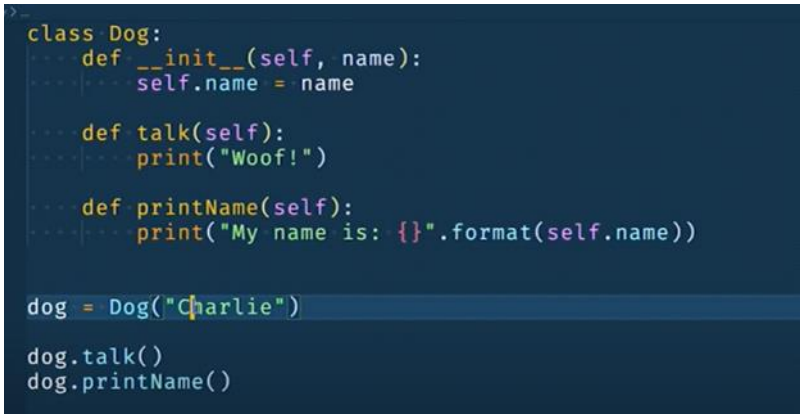


What is object && class


What are Objects and Classes?
01
Object is the basic unit of object-oriented programming
02
An object represents a particular instance of a class
03
There can be more than one instance of an object
04
Each instance of an object can hold its own relevant data
05
Objects with similar properties and methods are grouped together to form a Class

In it method


__init__() method in Python
Example
class Student(object):
def __init__(self, name, branch, year):
self.name = name
self.branch = branch
self.year = year
print("A student object is created.")
def print_details(self):
print("Name:", self.name)
print("Branch:", self.branch)
print("Year:", self.year)
ob1= Student("Paul","CSE",2019)
ob1.print_details()
A student object is created.
Name: Paul
Branch: CSE
Year: 2019
• __init__ is a special method in Python classes is a constructor method for a class
• __init__ is called when ever an object of the class is constructed

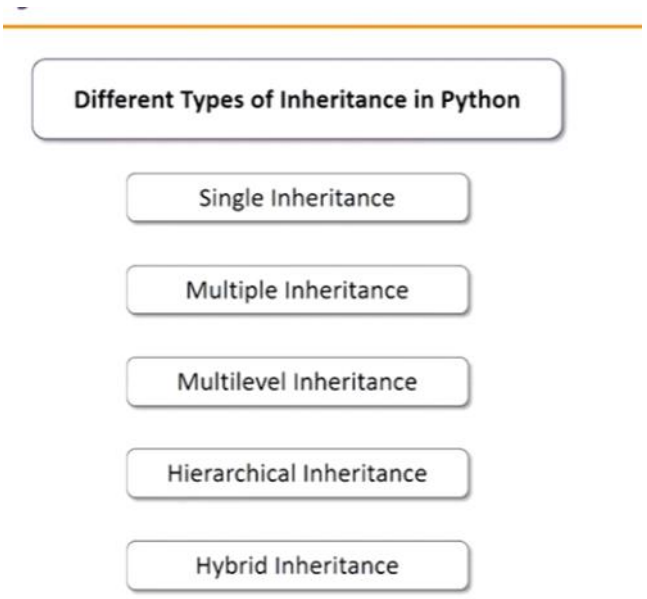


```
class Dog:
    def __init__(self, name):
        self.name = name

    def talk(self):
        print("Woof!")

    def printName(self):
        print("My name is: {}".format(self.name))


dog = Dog("Charlie")

dog.talk()
dog.printName()
```

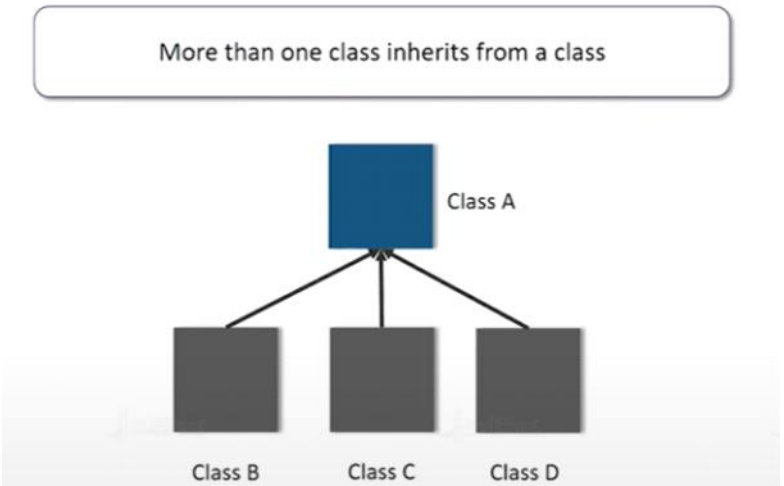
INHERITANCE

Friday, August 19, 2022 12:00 PM

Types of inheritance

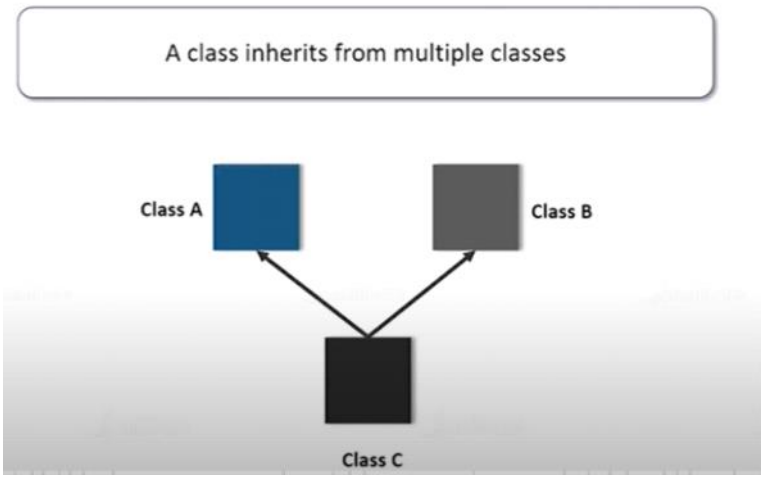


Hierarchical inheritance



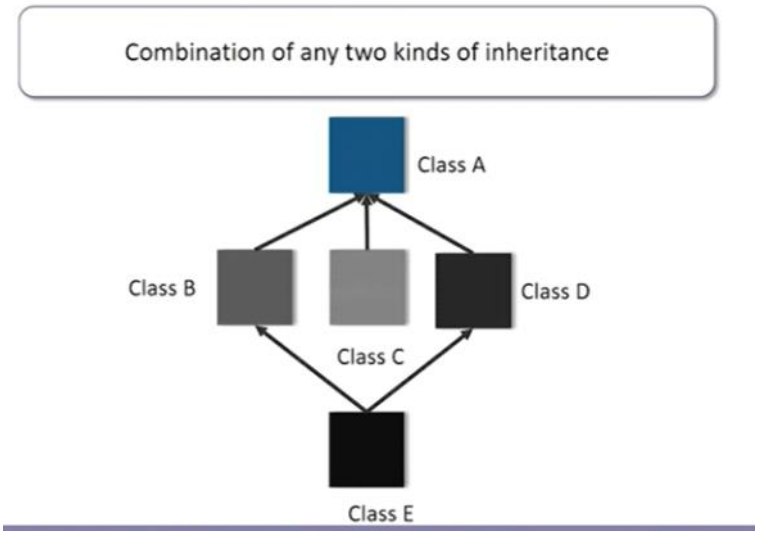
Here a class is acquiring things from multi classes

Multiple inheritance



Aa single class is giving his things to many classes

Hybrid inheritance

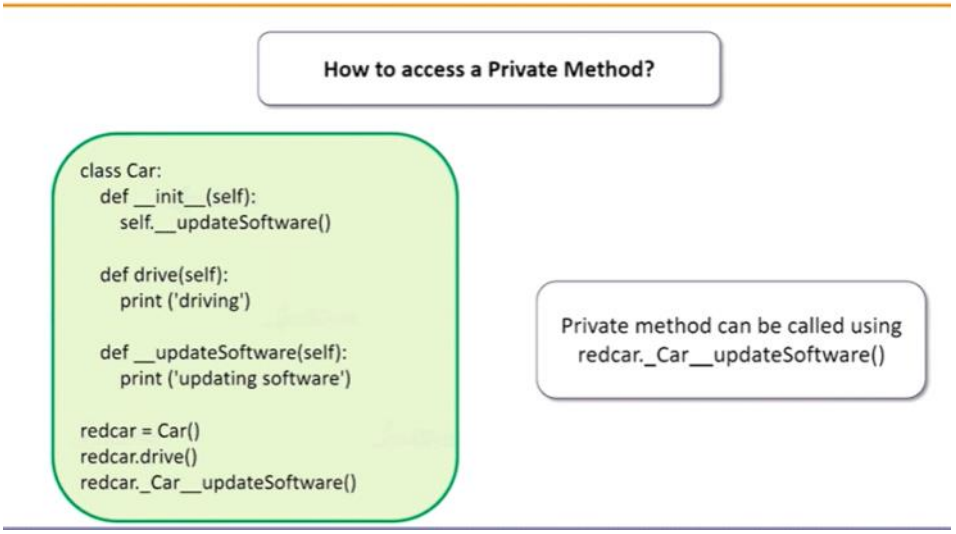


Here a class is giving to multiple classes and another class is taking things from multiple classes
Aa combination of
Multiple inheritance
Hierarchical inheritance

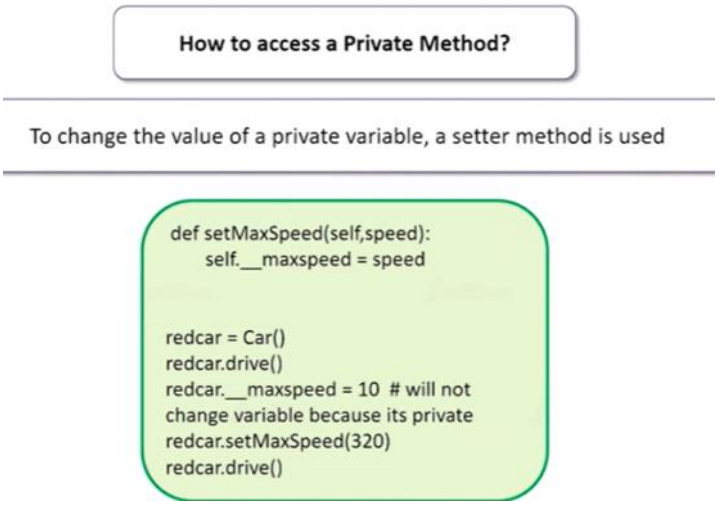
Encapsulation in Python

Saturday, August 20, 2022 11:04 AM

A



In python the private is describe by "__" two underscores but in it is A constructor not an private methods



So as we all know the private methods cannot be accessed by the user
So we will call it by class as a setter in a particular syntax "_class name__private method name ()"

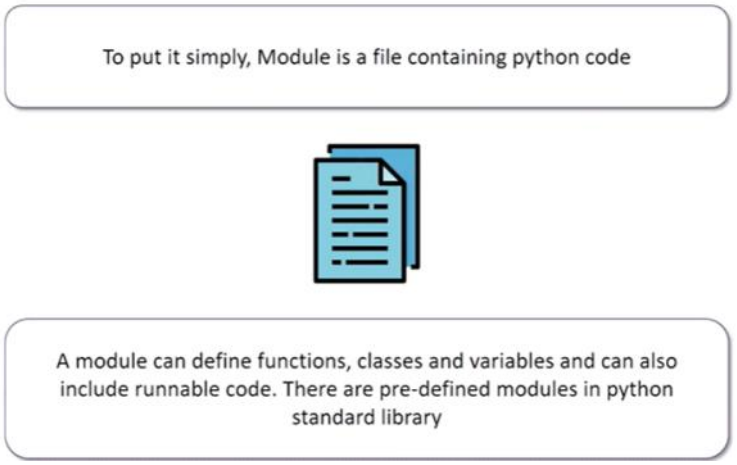
```python
class SomeClass:
    def public(self):
        print("Public Function")

    def __private(self):
        print("Private Function")

obj = SomeClass()

obj.public()
obj._SomeClass__private()

```

Here we get the output
Public Function
Private Function

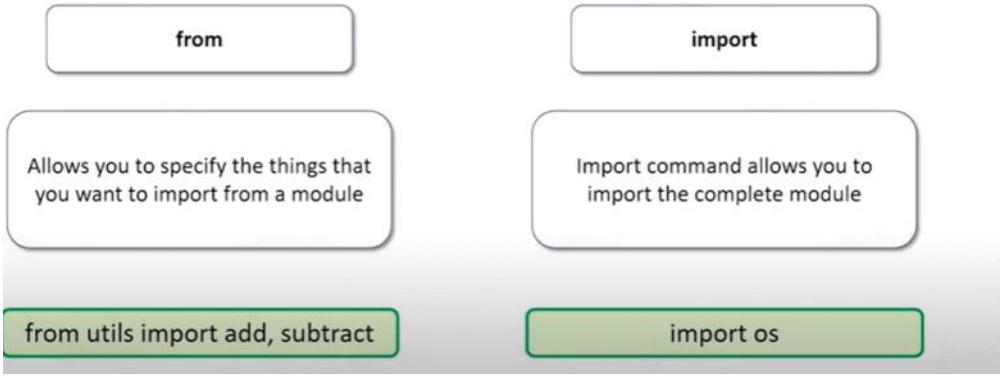
Modules in python

Saturday, August 20, 2022 11:32 AM

A

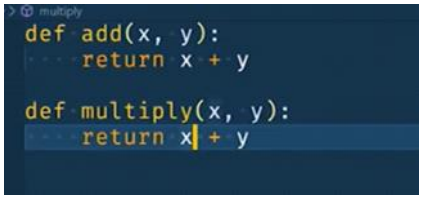


There two keywords which are important in the modules of python



Here they are majorly we will write a function in one file and by using from and import we can use that public method in another

File-1



```
def add(x, y):
    return x + y

def multiply(x, y):
    return x + y
```

File-2

```
1  from utils import add, multiply
2
3  print(add(7, 9))
```

We can also import the 2 functions

```
from utils import add, multiply

print(add(7, 9))
print(multiply(7, 9))
```

Standard libraries in the Python

Python's standard library is very extensive and offers a wide range of facilities

The standard library contains built-in modules, that provides access functionalities such as I/O. All the data types, functions and modules that we learned so far are available because of the standard library.

A

Here by using libraries of OS I can print any file data in my console in python

```
1  import os
2
3  print(os.listdir('./data'))
```

INSTALLATION OF PACKAGES

Saturday, August 20, 2022 11:45 AM

A

To install colorama
Open cmd write "pip install colorama"

By this we can the change the color of output in the command prompt

```
from colorama import init
from termcolor import colored

init()

print(colored("Hello, World!", "white", "on_red"))
```

Web scrapping

Saturday, August 20, 2022 11:58 AM

A

What is a web crawling and how does it work?

A web crawler works by **discovering URLs and reviewing and categorizing web pages**. Along the way, they find hyperlinks to other webpages and add them to the list of pages to crawl next. Web crawlers are smart and can determine the importance of each web page.

18-Feb-2022

What is Web Scraping?

Web scraping is an automatic method to obtain large amounts of data from websites. Most of this data is unstructured data in an HTML format which is then converted into structured data in a spreadsheet or a database so that it can be used in various applications. There are many different ways to perform web scraping to obtain data from websites. These include using online services, particular API's or even creating your code for web scraping from scratch. Many large websites, like Google, Twitter, Facebook, StackOverflow, etc. have API's that allow you to access their data in a structured format. This is the best option, but there are other sites that don't allow users to access large amounts of data in a structured form or they are simply not that technologically advanced. In that situation, it's best to use Web Scraping to scrape the website for data.

Web scraping requires two parts, namely the **crawler** and the **scraper**. The crawler is an artificial intelligence algorithm that browses the web to search for the particular data required by following the links across the internet. The scraper, on the other hand, is a specific tool created to extract data from the website. The design of the scraper can vary greatly according to the complexity and scope of the project so that it can quickly and accurately extract the data.