PYTHON SHORT NOTES

DATA TYPES IN PYTHON

String: **Sequence of characters, enclosed –“ ”/’ ’/””” “””, ordered, immutable, duplicates allowed**
Accessing Items: Indexing ,slicing and for loop
Functions: capitalize(), title(), upper(), lower(), lstrip() ,rstrip() ,strip(), swapcase(), replace(), find(), split(), join() ,endswith(), startswith(), del, index(), count(), isalnum(), isalpha(), isdecimal(), isdigit(), isnumeric(), islower() ,isupper(), istitle(), isspace(), zfill(),center().

List: **collection of heterogeneous datatype, enclosed –[ ], ordered, mutable duplicates allowed**
Accessing Items: Indexing ,slicing and for loop
Functions: append(), extend(), insert(), remove(), pop(), clear(), del, sort(), reverse(), index(), count()

Tuple: **collection of heterogeneous datatype, enclosed –( ), ordered, immutable, duplicates allow**
Accessing Items: Indexing ,slicing and for loop
Functions: Index(), count()

Set: **collection of immutable items, enclosed in –{ }.,unordered, mutable, duplicates not allows**
Accessing Items:for loop
Functions:remove(), discard(), pop(), clear(), del, add(), update(), union(), intersection(), intersection_update, difference() ,difference_update() ,symmetric_difference, symmetric_difference_update(), issubset(), issuperset(), isdisjoint(),
Frozen set: **collection of immutable items, enclosed in –{()}.,unordered, immutable, duplicates not allow.**
Accessing Items:for loop
Functions: Union(), intersection(), difference(), symmetric_difference.

Dictionary: **collection of key and value pairs, enclosed in –{ }., ordered, mutable, duplicates allows**
Keys:immutable,Values:mutables
Accessing Items: Indexing ,slicing and for loop
Functions: Dict_name[keyname], update(), pop(), popitems(), clear(), del, fromkeys(),setdefault().

Bool: True and False

FEATURES AND APPLICATION OF PYTHON

Key Features: Easy to learn and use,Expressive,Interpreted,cross platform,free and open source,OOp lang,vast library support,GUI support,Embeddable ,Dynamic

Applications: Web,software development, Business, Enterprise, Image processing, Game ,Audio & Video based,Console-based,Desktop GUI.

OPERATORS IN PYTHON

Arithmatic	(+) Addition,(-)Substraction,(*)Multiplication,(/)Division,(%)Modulas,(**)Exponents, (//)Floor Division
Assignment	(=) equal to,(+=) plus equal to,(-=) minus equal to,(*=) multiplication equal to , (/=) division equal, (//=) floor division equal to ,(**=) exponent equal to ,(%=) modulas equal to
Comparision	(==) equal to,(!=) not equal to,(<) less than,(>) greater than,(<=) less than equal to, (>=) greater than equal,
Membership	in:If present gives true otherwise false, not in: If not present gives true otherwise false
Logical	and: T&T=T otherwise F, or: F or F=F otherwise T,not:Negation
Identity	is:memory location same gives true, is not: memory location not same gives true
Bitwise	(&)bitwise and,(|)Bitwise or,(^)Bitwise xor,(>>)Right shift,(<<)left shift,(~)Bitwise not

LOOPS AND CONDITION STATEMENTS PYTHON

IF Else:	*if (condition): Statements else: Statements*	It is also called as decision making statement, condition is nessesary in if block. Else block is optional
IF Elif else:	*if (condition): Statements elif(condition): Statements else:*	We can apply multiple conditions using number of times elif statements
Nested IF else:	*if (condition): if(condition): . . . else: Statements else:*	We can use if inside if number of times and also we can apply condition inside conditions number of times
Short hand if:	*if (condition) : (statement)*	Use to avoid long code for single condtiton and satement in if block
Short hand if else:	*(State) if (condi) else (State)*	Use to avoid long code for single condtiton and when there is only one statement in if and else block
For loop:	*for var_name in (sequence): (conditions) (statements) …………………*	iterate,which means when we know how many times a statement has to be executed
Nested for loop:	*for var_name1 in (sequence): for var_name2 in (sequence2): (conditions) (statements) …………………*	We can use number of times for loops inside a for loop
While loop:	*While condition: (Statements) …….*	Use to iterations but we have to initialize and increment value manually
Nested while loop:	*While condition: While condition: (Statements) …….*	We can use number of times While loops inside a for loop
For else: While else:	*for var_name1 in (sequence): or While condition: (Statements) else: states*	After for/while block else block is executed
List comprehension:	*[expression for loop if condition]*	It provides a short syntax for creating a new list based on the values of an existing list.
Loop control state:	Break:It is use in only loop,it stop further lines of code and iterations Continue:Its stop current iterations and it continue with next iteratio	
Pass statement	There is no statement in loops or condition it use to avoid errors	

EXCEPTION HANDLING IN PYTHON

Buildin Exceptions	AttributeError,IndexError,ValueError,NameError,IndentationError, TypeError,KeyError,ZerodivisionError,OsError,Unicodeerror
try-except	*try:* *Code* *except:* *statements* Try:Test the block of code Except:It will handle the error

try-except-else	*try:*
	    Code
	except:
	   statements
	else:
	   statements
	else:If there is no error in try block then else block get execute

try-except-finally	*try:* *Code* *except:* *statements* *finally:* *statements* Finally: It get execute at every time after the execution of try or except block
traceback	It is library,it give detail information of error in else block
raise	We can pass our own massage to the error we can use in if else block

FUNCTIONS IN PYTHON

Function	It is a block of statements performing some specific task.
Built-in-function	--: The function which is already define in python
User define function	--: The function which is define by user
Function arguments	i)positional:When we are passing constants while calling function ii)Keyword: When we are passing variables with constants while calling function iii)Arbitary Positional:When we don't know how many argument as contants we are calling iv)Arbitary Keyword: When we don't know how many argument as variables with contants we are calling
return	When function find out return statement,it will exit the function scope
Lambda Function	It is anonymous function defining with lambda reserved keyword,We can perform small operation and expression should be one and we can pass number of arguments ,no need to use return,

SOME-BUILT IN FUNCTIONS PYTHON

zip()	use to create a dictionary from two iterables
filter()	It is a higher order function means we can pass function as first parameter it gives output in python object,it gives output for true
map()	It is a higher order function means we can pass function as first parameter it gives output in python object it gives actual output
all()	It gives true for all elements except 0 and false
any()	It gives true if any element is true.
chr()	It gives ASCII codes to charachters
Ord()	It coverts characters to ASCII.
bin()	It use to get binary value of numbers
bool()	It gives output in bool,for empty false otherwise true.
enumerate()	It gives index and values from sequence
range()	It gives numbers between given range
print()	It is use to print given statements
input()	It is use to receive an input from user.
eval()	It automatically identify input's datatype
abs()	It gives absolute value in numeric
dir()	It gives all functions from module
round()	It gives round off values upto given digit
iter()	It use to iterate with help of next keyword
pow()	It gives exponents of given number.
divmod()	It gives modulo value and complete answer
open()	It is use to open file

OOPs IN PYTHON

Object: It is physical entity,Every thing in python is an object.	
Method: It is function in say as method in class.	
Self: It is keyword and it is mandatory as first parameter in method, also use to access variable and methods inside the class.	
__init__ method: It is reserved method in python,it will get automatically executed when we create an object ,we can access parameters of the class in __init__method	
calling__init__ method in child class: We can call parent's __init__ method in child class by two ways 1)By Parent class name, 2)By super()	
1.Inheritance: Inheritance allows a class to inherit the properties from other class	
i)Single Inheritance	Child class is derived from only one parent class called single inheritance
ii)Multilevel Inheritance	The inheritance of a derived class from another derived class is called as multilevel inheritance
iii)Multiple Inheritance	When a class is derived from more than one base class it is called multiple Inheritance.
iv)Hierachical Inheritanve	When more than one classes are derived from one base class it is called Hierachical Inheritance
v)Hybrid Inheritance	It may be contain more than one type of inheritance
2.Polymorphism: We can use same function but with different signature into multiple classes	
3.Encapsulation: **Name mangling:**	Provide protection for data modification with the help of private modifiers. We can modify private method with the help of name mangling Hence we can say that python is not 100% OOPs lang.
4.Abstraction: First we need to inmport library ,Use to hide the internal functionality and we can also say that it can hide complexity of program with the help of *@abstractmethod*	
Overriding: when parent and child class contain same function name then parent class get overridden by child class .	
Overloading:when many classes contain same function but different parameters called as overloading.	

MODULE,PACKAGE AND FILE HANDLING

Module	Python file with extension .py is called module,it contains functions,variables and classes Important Keywords 1.import:use to import modules,libraries packages 2.from:We can import specific names from a module 3.as:Used to rename the module 4.(__name__=__main__):if in same module
Package	It is simply directory having collection of modules and __init__.py file
File Handling	It allows to user handle files. Modes:"w"-use to create file if not exist,if exist it overwrite,"r"-use to read file if exist ,if not gives error,"a"-use to create file if not exist ,if exist it add text,"x"-use to write file if exist it gives error.

LIBRARIES IN PYTHON

JSON	Json-Java Script Oriented Notation ,Front end developer always required data in JSON formate, json is nothing but the dictionary in python,dump():write file,load():read file,update():append file	
OS	Os-operating system.	
	1.os.rename()	:use to rename file
	2.os.remove()	:use to remove file
	3.os.path.exist()	:use to check file exist or not
	4.os.mkdir()	:use to create a directory
	5.os.makedirs()	:use to create nested directory
	6.os.rmdir()(empty file)	:use to remove directory,
	7.os.getcwd()	:use to get current directory path
	8.os.listdir()	:use to get list of files in directory
	9.os.path.join()	:use to join two paths
Glob	Glob stands for global,It used to find same extensions file using glob.glob()	
Shutil	It is use to copy file one location to another location by using shutil.copy() -Delete folder(containing file):shutil.rmtree("path")	
Time	It is use to reading current time and time complexity of program by using time.time()	
Date Time	It is grouping of date ,time,along its attributes year, month, day, hour, minute, second, microsecond. It is use to covert different date formate into standard formate.	
	1.datetime.today()	It gives todays date ,time ,year
	2.datetime.timedelta()	It differs date,time ,year,month, and week
	3.strptime()	It covert string to time object
	4.strftime()	It convert time object to string
	Strftime attributes:	%a - week day, %A -week day in full formate, %w -weekday as number ,%d -Day of month , %b -,Month in short ,%B – Month in full form , %m -Month of year ,%y -year in short , %Y-year in full form ,%P -AM/PM,%M -Minute, %S -second,%j -day of year , %U -week number of year
RE	Re-regular expression,used for searching and manipulating text	
	1.findall():It returns list of containing matches in the string Attributes: (\d)-[0-9], (\D)-[^0-9], (\w)-[a-zA-Z0-9], (\W)-except\w, (\s)-space, (\S)-except space, (\b)-boundary matches, (\B)-Except \b Meta Characters: (+)-one and more, (*)-zero or more, (.)-any one, (?)-zero and one, (^)-startswith, ($)-endswith,(|)-or 2.sub():multiple elements can be replace at time 3.search():It return object of first occurrence of match.subfunction:group()-return match, start()-start index ,end()-end index,span()-(start,end)index 4.match():It search at 0th index of match.subfunction:group()-return match, start()-start index ,end()-end index,span()-(start,end)index 5.complile():We can search match to multiple string 6.split():We can split string at the matches	

NUMPY IN PYTHON

1. np.array(list/tuple,ndmin=n)	:to create array with number of dim
2. array.ndim()	:returns dimensions of array
3. array.shape	:return shape of array
4. array.reshape(shape)	:use to reshape
5. np.zeros(shape)	:returns an array of zero values
6. np.ones(shape)	:returns an array of one values
7. np.arange(shape,size)	:similar to range
8 np.linspace(start,end,num=50)	:returns eventually spaced numbers over specified interval
9. np.eye(shape)	:return 2D array with one on diagonal and zero elsewhere
10. np.identity(n)	:return identity array
11. np.random:	:return random value array
rand()	:return random value between 0 and 1 interval array
randint(shape)	:return random value between given interval array
randn()	:return random normalize interval array
ranf()	:return random value between 0 and 1 interval array
random_sample()	:return random value between 0 and 1 interval array
12. np.sort(array,axis=0)	:return sorted array by ascending order
13. -np.sort(-array,axis=0)	:return sorted array by descending order
14. np.append(ar1,ar2)	:return extended array of two array
15. np.concatenate([a1,..,an])	:return concatenated array of number of array
16. np.nditer(array)	:It use in for loop to iterate each element of nD array
17. np.ndenumerate(array)	:return index and values after iteration
18. np.ceil(array)	:return closest and greatest integer.
19. np.floor(array)	:returns closest and lowest.
20. np.around(array,position)	:returns round off upto given decimal
21. np.where(array=value)	:returns index of given element.
22. np.argmax(array,axis=none)	:returns indices of the maximum value
23. np.delete(array,obj,axis=none)	:use to delete column and rows
24. np.max(array)	:return max value of array
25. np.min(array)	:return min value of array
26. np.maximum(ar1,ar2)	:return max array value from two array
27. np.minimum(ar1,ar2)	:return min array value from two array
28. np.square(array)	:return square value of array
29. np.sqrt(array)	:return square root value of array
30. np.cbrt(array)	:return cube root value of array
31. array.flatten()	:convert nD array to 1D array
32. array.tolist()	:convert array to list
33. array.ravel()	: convert nD array to 1D array and return refferece original
34. array.copy()	: return deep copy of given array
35. np.full(shape,value)	:return array of given same value
36. np.power(array,n)	:return power of array
37. np.percentile(array)	:return percentile of given array

Basic mathematical	
1. np.add(arr1,arr2)	:return element wise addition of two array
2. np.multiply(arr1,arr2)	:return element wise multi of two array
3. np.dot(arr1,arr2)	:return dot product of two array
4. np.subtract(arr1,arr2)	:return element wise substaction of two array
5.np.divide(arr1,arr2)	:return element wise division of two array
Statistical	
1. np.mean(array) # No outliers	:Return mean of array and use when no outliers
2. np.median(array) # outliers	:Return median of array and use when outliers
3. np.std(array)	:Return standard deviation of array
4. np.var(array)	:Return varience of array .
Linear Algebric	
1.np.linalg.solve(A,B)	:Return solution of linear equation of array
2. np.linalg.inv(A)	:Return inverse of array
3. np.linalg.det(A)	:Return determinant of array
Logarithmic	
1.np.log(array)	:Return solution of natural log
2. np.log2(array)	:Return solution of base 2 log
3. np.log10(array)	:Return solution of base 10 log
Trignometric	
1. np.deg2rad(array)	:covert degree to radian value
2. np.rad2deg(array)	:covert radian to degree value
3. np.sin()	:return trigonometric angle of sin
4. np.cos()	:return trigonometric angle of cos
5. np.tan()	:return trigonometric angle of tan
6. np.pi	:return value of pi.