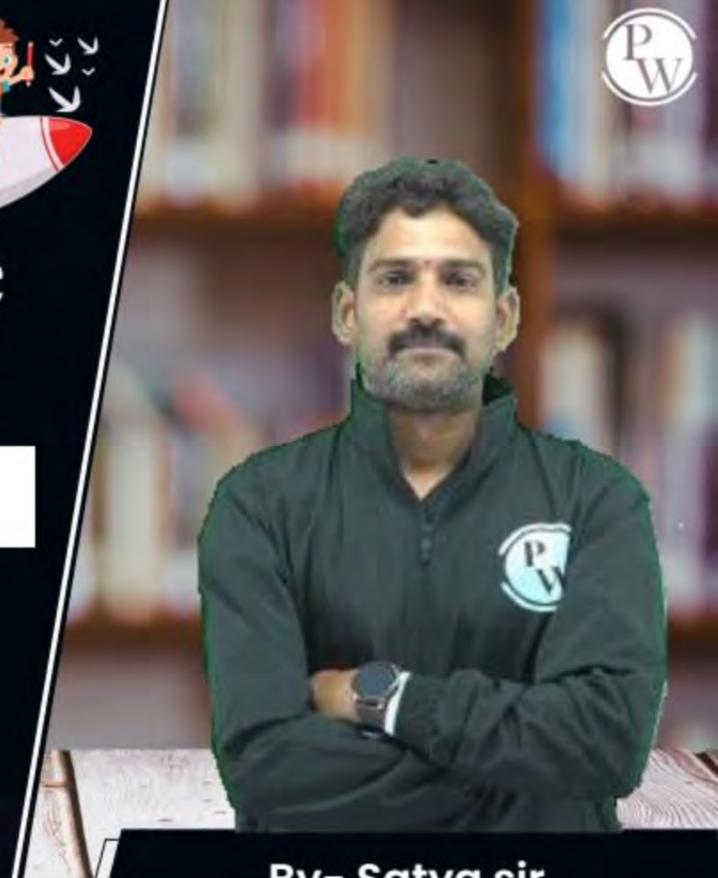
Data Science & Artificial Intelligence

Python For Data Science

Classes & Modules



By- Satya sir

Recap of Previous Lecture









- Dictionaries
- Functions
 - Recursion

Topics to be Covered









- Method overloading
- method oversiding
- Gnstructor
 - Modules
 - math
 - date time
 - numpy





Method overloading:

- The oubility to define two or more methods with same name but difference in the Arguments [with in the same class]

Method overloading.

ob. x(5,7) # 12

06. x (4, 5, 6) # 120



Method-Over riding

: The orbility to define method with same Name and Signature (arguments) in Pavent class and child class as known as method-oversiding.

class A: def M1(): Print (Hello) Class B(A): dy M2(): overside Brist (' Bye') def M1(): Print (Hail) obe B() 06. NI() # Have

06.M2() # Bye

Method

- defined within class

- Object . method (arguments)

- At a time only one object (an be used with method

x.y.z.MI()

Snvalid

X.MI()

Y.MI()

Valid

Z.MI()

function

defined outside class

- function (objects)

- Multiple objects can be arguments for a dunction.

Ex: function (x, y, 3)

X=4

def f(a,b,c):

Y=2.7

Print(a,b,c)

\$=1 (ATE)

f(x,y,z)

object of these clay in the set





```
Super() Statement: To ouccess overridden members of super class.
class A:
                              ob=B()
 >def MI():
                              ob.MI() # Hello # Super(B, Ob).MI()
      Print (' Hello')
                                         # Hai
                               06.M2() # Byl
class B(A):
    def MI ():
      _Super(). MI()
       Rint ( Hai')
     del W5():
         Print (18x1)
```

Constructor: It is also as method, in voked | Called | Executed while Instance | Object Creation, automatically

Lyso, It is called Initialization method





args, Kwargs => Both are used to Pars Variable No. of arguments to a function one thod.

Skepwird arguments

Ex: def f(** Kwargs):

def f(** Kwargs): for key, value in Kwargs. Items. Point ('/s=/s), /. (Key, value)) f({a':3,b':4, ":5})

Access Modifiers

1) Public: By default Every member is Public.

2) Brotected: _ (single underscore) Rreflex, it is Protected member. It can be accessed in only it's inherited classes.

3) Private: - Any member with __ (double undexscore) Prefix "is Private member.

au=10 #Public - Private members are acceptible only with in despective class by it's own object only. class B(A):

_b=20 # Protected __C=30 # Private

def M():

Print (a, b, -- C, In class A')

10 20 30 in chu A

dy M1():

Print (a) # 10

Print (Super(). _b) # 20

Point (Super()._c)

Error

dy M2():

Print(a) #10

Print (b) # Error

Print (--c) # Error

Print (Ob. _b) # Valid Print (Ob. _c) # Invalid

Module:

- Any or Every Program in Bython is called a Module.
- 2 Types of modules:
 - 1) Bredefined modules: detetime, math, os, sys, collection, numpy ---
 - 2) User defined modules: Every Python Program saved as . Py Extension.
- To use avail services of one module an another, import keyword as used

import Modulename

- (OR) import Package * (All modules in Package)
- (OR) from module support clans

 Ex: from most support albs



Print (Strftime (Harrective))

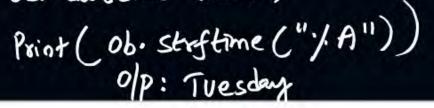


Directive	Description	Example
%a	Weekday, short version	Wed
%A	Weekday, full version	Wednesday
%w	Weekday as a number 0-6, 0 is Sunday	3
%d	Day of month 01-31	31
%b	Month name, short version	Dec
%B	Month name, full version	December
%m	Month as a number 01-12	12
%y	Year, short version, without century	18
%Y	Year, full version	2018
%H	Hour 00-23	17
%1	Hour 00-12	05
%р	AM/PM	PM
%M	Minute 00-59	41
%S	Second 00-59	08
%f	Microsecond 000000-999999	548513



import daletime

obs daletime. now()



		Il. Ingany
Directive	Description	Example
%z	UTC offset	+0100
%Z	Timezone	CST
%j	Day number of year 001-366	365
%U	Week number of year, Sunday as the first day of week, 00-53	52
%W	Week number of year, Monday as the first day of week, 00-53	52
%с	Local version of date and time	Mon Dec 31 17:41:00 2018
%C	Century	20
%x	Local version of date	12/31/18
%X	Local version of time	17:41:00
%%	A % character	%
%G	ISO 8601 year	2018
%u	ISO 8601 weekday (1-7)	1
%V	ISO 8601 weeknumber (01-53)	01







Method	Description	
math.acos()	Returns the arc cosine of a number	
math.acosh()	Returns the inverse hyperbolic cosine of a number	
math.asin()	Returns the arc sine of a number	
math.asinh()	Returns the inverse hyperbolic sine of a number	
math.atan()	Returns the arc tangent of a number in radians	
math.atan2()	Returns the arc tangent of y/x in radians	
math.atanh()	Returns the inverse hyperbolic tangent of a number	
math.ceil()	Rounds a number up to the nearest integer	
math.comb()	Returns the number of ways to choose k items from n items without repetition and order	
math.copysign()	Returns a float consisting of the value of the first parameter and the sign of the second parameter	
math.cos()	Returns the cosine of a number	
math.cosh()	Returns the hyperbolic cosine of a number	
math.degrees()	Converts an angle from radians to degrees	
math.dist()	Returns the Euclidean distance between two points (p and q), where p and q are the coordinates of that point	
math.erf()	Returns the error function of a number	
math.erfc()	Returns the complementary error function of a number	





Method	Description
math.exp()	Returns E raised to the power of x
math.expm1()	Returns E ^x - 1
math.fabs()	Returns the absolute value of a number
math.factorial()	Returns the factorial of a number
math.floor()	Rounds a number down to the nearest integer
math.fmod()	Returns the remainder of x/y
math.frexp()	Returns the mantissa and the exponent, of a specified number
math.fsum()	Returns the sum of all items in any iterable (tuples, arrays, lists, etc.)
math.gamma()	Returns the gamma function at x
math.gcd()	Returns the greatest common divisor of two integers
math.hypot()	Returns the Euclidean norm
math.isclose()	Checks whether two values are close to each other, or not
math.isfinite()	Checks whether a number is finite or not





Method	Description	
math.isinf()	Checks whether a number is infinite or not	
math.isnan()	Checks whether a value is NaN (not a number) or not	
math.isqrt()	Rounds a square root number downwards to the nearest integer	
math.ldexp()	Returns the inverse of math.frexp() which is x * (2**i) of the given numbers x and i	
math.lgamma()	Returns the log gamma value of x	
math.log()	Returns the natural logarithm of a number, or the logarithm of number to base	
math.log10()	Returns the base-10 logarithm of x	
math.log1p()	Returns the natural logarithm of 1+x	
math.log2()	Returns the base-2 logarithm of x	
math.perm()	Returns the number of ways to choose k items from n items with order and without repetition	
math.pow()	Returns the value of x to the power of y	
math.prod()	Returns the product of all the elements in an iterable	
math.radians()	Converts a degree value into radians	





Method	Description	
math.isinf()	Checks whether a number is infinite or not	
math.isnan()	Checks whether a value is NaN (not a number) or not	
math.isqrt()	Rounds a square root number downwards to the nearest integer	
math.ldexp()	Returns the inverse of math.frexp() which is x * (2**i) of the given numbers x and i	
math.lgamma()	Returns the log gamma value of x	
math.log()	Returns the natural logarithm of a number, or the logarithm of number to base	
math.log10()	Returns the base-10 logarithm of x	
math.log1p()	Returns the natural logarithm of 1+x	
math.log2()	Returns the base-2 logarithm of x	
math.perm()	Returns the number of ways to choose k items from n items with order and without repetition	
math.pow()	Returns the value of x to the power of y	
math.prod()	Returns the product of all the elements in an iterable	
math.radians()	Converts a degree value into radians	





Constant	Description
math.e	Returns Euler's number (2.7182)
math.inf	Returns a floating-point positive infinity
math.nan	Returns a floating-point NaN (Not a Number) value
math.pi	Returns PI (3.1415)
math.tau	Returns tau (6.2831)





- NumPy is a Python library used for working with arrays.
- It also has functions for working in domain of linear algebra, fourier transform, and matrices.
- NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely.
- NumPy stands for Numerical Python.
- NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.
- The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy.



2 mins Summary



- -Method overloading
- method oversiding
- Constructors
- Access modifiers
- modules datetime

- math, numpy (please refer the document)

To be contd...



THANK - YOU