1. Vars are declared in python using words  
   one = 1203;
2. Control structures
   1. if condition:  
      do something;  
      else:  
      do something
   2. elif  
      if condition:  
      do something  
      elif cond:  
      do something  
      else:  
      do something
3. while and for
   1. while loop\_condition:  
      do something;  
      alter loop\_condition;
   2. for var in sequence:  
       for body
4. array and list
   1. mathematical functions can be performed on an array
   2. items can be added in a list using append() : list\_name.append(items to be appended)
   3. list are declared using [] ;
   4. list.pop(item ->remove)
   5. tuple and list are similar except tuples are immutable whereas lists are mutable tuples are declared using tuple =();
   6. dictionary : collection of key value pairs dictionary = {};
5. looping through ds
   1. iteration through list is done using for
   2. dictionary  
      for key in dictionary:  
      print(‘%s 🡪 %s’ %(key,dictionary[key]));
   3. using iteritem()  
      for key,val in dictionary.items():  
      print(‘my %s is %s‘ %(key, val));
6. class and objects
   1. objects has data(attributes) and behavior(methods)
   2. class is a blueprint from which objects are created
   3. pass statement is used when a statement is required syntactically but execution of any code is not required  
      class vehicle:  
      pass
   4. objects are instances of class  
      car = vehicle();
   5. class Vehicle:  
      def \_\_init\_\_(self, number\_of\_wheels,type\_of\_tank,max\_velocity,seating\_capacity):  
      self.number\_of\_wheels = number\_of\_wheels

self.type\_of\_tank = type\_of\_tank

self.max\_velocity = max\_valocity  
self.seating\_capacity = seating\_capacity

tesla\_mpodel\_s = Vehicle(4, “electric”, 300, 2)

def number\_of\_wheels(self):

return self.number\_of\_wheels

def set\_number\_of\_wheels(self, number):

self.number\_of\_wheels = number

#getters are defined by @property  
@property

def number\_of\_wheels(self):

return self.number\_of\_wheels

#setters are defined by @property\_name.setter

@number\_of\_wheels.setter

Def number\_of\_wheels(self, number):

Self.number\_of\_wheels = number

#we can use these methods as attributes

Tesla = Vehicle(4, “electric”,300,2)

Print(Tesla.number\_of\_wheels); #getter

Tesla.number\_of\_wheels = 10 #setter

1. encapsulation
   1. it is a method that restricts direct access to objects data and methods
   2. public instance variable  
      if public instance variable are initialized in constructor method then it has to be provided during object initialization  
      class Person:  
      def \_\_init\_\_(self, age):  
      self.age=age  
        
      debo =Person(27)  
        
      public instance vars can also be declared in class  
      class Person:  
      first\_name = “debo”  
        
      #we do not need to provide the value during object initialization as all objects will have the variable with default value during initialization. The default value can be mutated at later stage.  
      Employee = Person();  
      employee.first\_name = “Debopriyo”
   3. non public instance variable declaration and utilization is same as public instance variable except non public instance vars are declared using \_(underscore) prepended to their name  
      non public instance variables are just convention and can be updated
   4. in python methods, the object itself is passed on as first arg
   5. to avoid passing “self” , a method has to be declared as static method using @staticmethod decorator
   6. \_\_init\_\_ is not a constructor as a reference to the object is passed. \_\_init\_\_ is called after the object is created and is used to initialize it
2. Inheritance
   1. We pass parent class as parameter to the child class to inherit the properties  
      class ElectricCar(Vehicle):  
      def \_\_init\_\_(self, number\_of\_wheels, type\_of\_tank,max\_velocity,seating\_capacity):  
      Vehicle.\_\_init\_\_( self, number\_of\_wheels, type\_of\_tank,max\_velocity,seating\_capacity)