Python Training.....

-- Jeetendra Bhattad

MetaClasses

- Classes in Python are also Objects
- Metaclass is an object that knows how to create and manage class
- e.g class Foo(object): pass
  - isinstance(Foo, object) – returns true
  - type(Foo) – returns <type 'type'>
- When new class is defined with the class statement following things happen
  - Body of class is executed as series of statements
  - Its own dictionary is generated
  - Name mangling happen on private members
  - Finally, name of the class, list of base classes and dictionary are passed to the constructor of metaclass to create the corresponding class object.
  - e.g refer create_class.py
  - One of the use-case : create database classes at runtime.

MetaClasses

- MetaClasses are nothing but a way of applying Class Decorators with following advantages
  - MetaClasses are applied when Class is created
  - MetaClasses are applied to the complete inheritance heirarchy unlike decorators
- Examples : i) class_decorator_to_check.py for verifying whether class has manager methods.
  ii) simple_meta_class.py : metaclass for verifying whether class has manager methods.
  iii) meta_class_inheritance.py : explains how meta-class are applied to complete inheritance heirarchy and not just base class.