NPRG065: Programming in Python Lecture 11

http://d3s.mff.cuni.cz



Tomas Bures

Petr Hnetynka

{bures,hnetynka}@d3s.mff.cuni.cz



CHARLES UNIVERSITY IN PRAGUE faculty of mathematics and physics

setattr__(self, name, value)

called when an attribute assignment is attempted

getattr__(self, name)

- called when the default attribute access fails with an AttributeError
 - if an attribute already has a value, ___getattr___() is not used
- delattr__(self, name)
 - like __setattr__() but for attribute deletion instead of assignment

____dir___(self)

- called when dir() is called on the object
 - a sequence must be returned



dir(object)

- if the object has a method named _____dir___(), this method will be called
- if the object does not provide _____dir___(), the function tries to gather information from the object's _____dict____ attribute, if defined, and from its type object

___dict

- a dictionary used to store an object's attributes
- created automatically

- Why use ______()
 - creating immutable objects (with __slots__)
 - lazy creation of attribute values
 - own property-like behavior but in a single method
 - creating attributes when setting values to other attributes



_____slots

- a class variable that can be assigned a string, iterable, or sequence of strings with variable names used by instances
- reserves space for the declared variables
- prevents the automatic creation of ___dict___
- i.e., no other than declared variables can be created
- plus objects with slots are smaller and faster
 there is no dynamic dict
 - See slots.py

__getattribute__(self, name)

- called unconditionally to attribute accesses
 - default implementation locates value in ___dict__ or ___slots___
- if the class also defines __getattr__(), the latter will not be called unless __getattribute__() either calls it explicitly or raises an AttributeError
- to access other attributes from <u>getattribute</u>, call the base class method (to avoid recursion)

• object.__getattribute__(self, name)

- usages
 - preventing access to attributes
 - inventing new attributes (like with __getattr__ but without look for existing attributes)
- rarely used

6

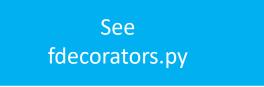
attribute.py

Function decorators

```
@decorate
def function():
    pass
```

- decorator ~ a function modifying a function to create new function
 - code above is equivalent to

```
def function():
    pass
function = decorate( function )
```



Function decorators

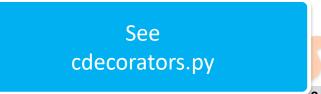
- many predefined decorators
 - @property, @staticmethod
 - we already know them
 - functools module
 - many useful decorators (not only for defining other decorators)



Class decorators

- Similar to function decorators
- A function receiving a class object as an argument and returning a class object as a result

Less commonly used than function decorators







Department of Distributed and Dependable 0-0