Python in Web Development

Recap

- Python 2 vs 3
- Basic constructs
- Data Structures
- Builtin modules

© Copyright 2017. 3Pillar | All rights reserved

LAB 3

OOP in Python

© Copyright 2017. 3Pillar | All rights reserved

AGENDA

- Definitions and principles
- Inheritance
- Encapsulation
- Polymorphism

© Copyright 2017. 3Pillar | All rights reserved

Classes

```
class Student(object):
    pass
```

© Copyright 2017. 3Pillar | All rights reserved

Class variables

```
class Student(object):
    # class variable shared by all instances
    faculty = “informatics”
```

© Copyright 2017. 3Pillar | All rights reserved

Instance variables

```
class Student(object):
    def __init__(self, age):
        # instance variable unique to each instance
        self.age = age

s = Student(20)
print(s.age)
>>> 20
```

© Copyright 2017. 3Pillar | All rights reserved

Inheritance

```
class Student(object):
    pass

class MasterStudent(Student)
    pass
```

© Copyright 2017. 3Pillar | All rights reserved

Multiple Inheritance

```
class Human(object):
    pass

class Computer(object)
    pass

class Cyborg(Human, Computer)
    pass
```

© Copyright 2017. 3Pillar | All rights reserved

Encapsulation

- by convention
- *_varname*
- *__varname* replaced with *_classname__varname* ()

© Copyright 2017. 3Pillar | All rights reserved

Encapsulation @property

```
class Student(object):
    def __init__(self):
        self._age = 0

    @property
    def age(self):
        return self._age

s = Student()
print(s.age)
>>> 0
```

© Copyright 2017. 3Pillar | All rights reserved

Encapsulation @*property*.setter

```
class Student(object):
   def __init__(self):
      self._age = 0

   @property
   def age(self):
      return self._age

   @age.setter
   def age(self, age):
      self._age = age

s = Student()
print(s.age)
>>> 0
s.age = 20
print(s.age)
>>> 20
```

© Copyright 2017. 3Pillar | All rights reserved

Polymorphism

```
class Person(object):
   def learn(self):
      raise NotImplementedError("Implement in subclass")

class Student(Person):
   def learn(self):
      print("learning to party")

>>> for s in [Student(), Programmer()]:
...    s.learn()
learning to party
learning to work
```

© Copyright 2017. 3Pillar | All rights reserved

Polymorphism

```
class Person(object):
   def eat(self):
      print("Healthy food")

class Programmer(Person):
   def eat(self):
      super(Programmers, self).eat()
      print("Pizza")

>>> for s in [Person(), Programmer()]:
...    s.eat()
>>> Healthy food
>>> Healthy food
>>> Pizza
```

© Copyright 2017. 3Pillar | All rights reserved

Problems

Apply the concepts you learned

© Copyright 2017. 3Pillar | All rights reserved

Resources

- Python property
- Python Classes
- Python's Class Development Toolkit (video)

© Copyright 2017. 3Pillar | All rights reserved

#1 Given the year of birth, calculate a person's age using a class

Extra points for:

- get the full birth date as string("dd/mm/yyyy"), calculate the **age as x years, y months and z days** old

© Copyright 2017. 3Pillar | All rights reserved

#2 Extend Shape to implement methods that compute the area for a Circle and a Square

© Copyright 2017. 3Pillar | All rights reserved

#3 Implement the TicTacToe game (X si Zero)

© Copyright 2017. 3Pillar | All rights reserved

Quiz time :)

https://goo.gl/forms/mk96emymdznPhh8T2

© Copyright 2017. 3Pillar | All rights reserved

Homework

- Learn Python the Hard Way, ex. 40 - 44

© Copyright 2017. 3Pillar | All rights reserved

Thank you!

© Copyright 2017. 3Pillar | All rights reserved