UNIVERSITÄT BIELEFELD
Faculty of Technology

Programming

Object-oriented programming

Luna Pianesi

Faculty of Technology, Bielefeld University

Loops

Functions

Classes, Modules & Packages

Programming Errors & Debugging

Creating new types

- A `class` defines a new type
- It can provide
  - class variables & functions
  - instance variables & functions

Classes—example of code reuse

```python
class Library:
    description = 'This␣is␣a␣Library'

    def __init__(self, name):
        # name the library
        self.name = name
        # create empty book storage on initialization
        self.storage = list()

    def addBook(self, book):
        self.storage.append(book)

    def getAllBooks(self):
        return tuple(self.storage)

myLib = Library('Bodleian␣Library')
myLib.addBook('The␣Art␣of␣Computer␣Programming␣(D.␣Knuth)')
```

UNIVERSITÄT BIELEFELD

Modules

- Every `.py` file is a module
- Modules can host functions, variables, and classes
- Imported modules with `import` statement
- Should not have blocks of code that are immediately executed
- Explicit reference to module scope: `global`
- Name of module available as global variable `__name__`

UNIVERSITÄT
BIELEFELD

Modules—example of code reuse

mystringutils.py

```
#
# A module for all kinds of string utils
#

def findSubstringInStrings(stringCollection,
     pattern):
    occ = list()
    for i, s in enumerate(stringCollection):
        j = s.find(pattern)
        while j != -1:
            occ.append((i, j))
            j = s.find(pattern, j+1)
    return occ
```

myscript.py

```
#!/usr/bin/env python3

import mystringutils

if __name__ == '__main__':
    myStringList = ['the␣rain␣in␣spain',
        'ain\'t␣no␣sunshine',
        'she␣was␣greeted␣with␣disdain']

    occOfAin = mystringutils.
        findSubstringInStrings(myStringList,
         'ain')
    print(occOfAin)
```

Modules—example of code reuse

mystringutils.py

```python
#
# A module for all kinds of string utils
#

def findSubstringInStrings(stringCollection,
      pattern):
    occ = list()
    for i, s in enumerate(stringCollection):
        j = s.find(pattern)
        while j != -1:
            occ.append((i, j))
            j = s.find(pattern, j+1)
    return occ
```

myscript.py

```python
#!/usr/bin/env python3

import mystringutils as su

if __name__ == '__main__':
    myStringList = ['the␣rain␣in␣spain',
        'ain\'t␣no␣sunshine',
        'she␣was␣greeted␣with␣disdain']

    occOfAin = su.findSubstringInStrings(
        myStringList, 'ain')
    print(occOfAin)
```

UNIVERSITÄT BIELEFELD

Modules—example of code reuse

mystringutils.py

```python
#
# A module for all kinds of string utils
#

def findSubstringInStrings(stringCollection,
     pattern):
    occ = list()
    for i, s in enumerate(stringCollection):
        j = s.find(pattern)
        while j != -1:
            occ.append((i, j))
            j = s.find(pattern, j+1)
    return occ
```

myscript.py

```python
#!/usr/bin/env python3

from mystringutils import
    findSubstringInStrings

if __name__ == '__main__':
    myStringList = ['the␣rain␣in␣spain',
        'ain\'t␣no␣sunshine',
        'she␣was␣greeted␣with␣disdain']

    occOfAin = findSubstringInStrings(
        myStringList, 'ain')
    print(occOfAin)
```

Packages

- Way of structuring multiple modules into a directory hierarchy
- Package directories must contain a `__init__.py` file
- Can be imported the same way as modules
- Python itself offers many packages, and even more third-party packages are available through *package managers* such as `conda`

Quiz

- In Python, a class is ____________ for an object.

  a nuisance　　an instance　　a blueprint　　a distraction

- Consider the following class:

```python
class Dog:
    def __init__(self, name, age):
        self.name = name
        self.age = age
```

  What is the correct statement to instantiate a `Dog` object?

  - `Dog('Rufus', 3)`
  - `Dog(self, 'Rufus', 3)`
  - `Dog.__init__('Rufus', 3)`

source (in part): `https://realpython.com/quizzes`

UNIVERSITÄT
BIELEFELD

Quiz

- In Python, a class is _____________ for an object.

  a nuisance   an instance   a blueprint ✓   a distraction

- Consider the following class:

```python
class Dog:
    def __init__(self, name, age):
        self.name = name
        self.age = age
```

What is the correct statement to instantiate a `Dog` object?

  - `Dog('Rufus', 3)` ✓
  - `Dog(self, 'Rufus', 3)`
  - `Dog.__init__('Rufus', 3)`

source (in part): `https://realpython.com/quizzes`

Recap

Summary

- Code reuse through
    - Classes
    - Modules & Packages

What comes next?

- Write your first classes and modules
- Due date for this week's exercises is ***Wednesday, Dec 6, 2pm, 2023***.

Next lecture: Input, file processing & text mining ...