FOLDABLE ROBOTICS

Class V: Object-Oriented Programming

Python

- Anaconda
- Basic layout
- types: int, float, string, tuple, list, dict, set, etc
- If, for, while
- Indexing
- Functions
- Documentation
- packages

IDEAB

Thousands of Tutorials…

- So many that I don’t need to make my own for the basics
- https://wiki.python.org/moin/BeginnersGuide/Programmers

IDEAB

Matlab-like

- Scripting and/or Structure
- Numpy, Scipy & Matplotlib
  - OpenCV, Sympy, and many more.
- https://docs.scipy.org/doc/numpy-dev/user/numpy-for-matlab-users.html
- http://matplotlib.org/users/pyplot_tutorial.html

IDEAB

Spyder vs Jupyter

- Debugging
- Visualization
- Ease of use

- Thoughts?
- I suggest spyder for object-oriented code

IDEAB

Pythonic coding

- Everything is accessible, for better or worse
- Conventions for signaling private data or obfuscating it

IDEAB

Try…

```
import this
```

IDEAB

The Zen of Python

by Tim Peters

Beautiful is better than ugly.

Explicit is better than implicit.

Simple is better than complex.

Complex is better than complicated.

Flat is better than nested.

Sparse is better than dense.

Readability counts.

Special cases aren't special enough to break the rules.

Although practicality beats purity.

Errors should never pass silently.

Unless explicitly silenced.

In the face of ambiguity, refuse the temptation to guess.

There should be one-- and preferably only one --obvious way to do it.

Although that way may not be obvious at first unless you're Dutch.

Now is better than never.

Although never is often better than *right* now.

If the implementation is hard to explain, it's a bad idea.

If the implementation is easy to explain, it may be a good idea.

Namespaces are one honking great idea -- let's do more of those!

https://www.python.org/dev/peps/pep-0020/

IDEAB

Coding styles in Python

- Object-oriented
- Imperative
- Procedural
- Functional

https://blog.newrelic.com/2015/04/01/python-programming-styles/

- Others
  - Structured
  - Event-driven
  - Declarative

IDEAB

Object-oriented coding style

- Reusability
- Modularity
- Inheritability
- Break a problem down by structure vs
- Break a problem down by function
- Bundle function and data together.

IDEAB

Classes

- Contains all the information needed to use that concept
  - Data
  - Methods(Functions)
  - Initialization / Garbage collection
- Hides complexity from user of the class

IDEAB

Functions, Classes, etc

- my_function
- my_function(arguments)
- instance=MyClass()
- instance.method(arguments)
- instance.data
- instance.property

IDEAB

Inheritance

- Make generic objects and specify as needed
- Can inherit from multiple sources

IDEAB

Overloading

- Operator Overloading
  - Use the same operator for vastly different functionality
  - Is integer multiplication the same as float?
- Function overloading
  - Same function name, different arguments or operations

```
class MyClass(object):
    def
```

IDEAB

Operations & Operators

- Name common operators in Python

IDEAB

Operations & Operators

- Name common operators in Python
  - + - / * % ** //
  - & | ^ ~ << >>
  - == != < > <= >=
  - += -= *= /= %=
  - not, and, or
  - is, is not, in, not in

IDEAB

Functions

- def function(*args, **kwargs):
  - Arguments
  - Keyword arguments
  - Default values
- https://docs.python.org/3/library/functions.html

IDEAB

Scoping

- Can read variables declared in larger scopes, but cannot write by default
- Anything created within its scope is destroyed upon leaving the function
- Global vs local variables

```
a = 0

def my_function():
    a = 3
    print(a)

my_function()

print(a)
```

http://python-textbok.readthedocs.io/en/latest/Variables_and_Scope.html

IDEAB

Emulating Types

- https://docs.python.org/3/reference/datamodel.html#emulating-container-types
- https://docs.python.org/3/reference/datamodel.html#emulating-numeric-types

IDEAB

Packages

IDEAB

Shapely

- http://toblerity.org/shapely/manual.html
- C++ library access
- Data stored in library objects, not native python

IDEAB

Shapely Exercises

- What is the result of an operation
- How do you access coordinates
- How do you access geometries
- How do you transform geometries
- How do you apply functions
- How do you plot geometries?

IDEAB

Recursion

- How do you flatten nested lists of lists?

IDEAB

3D Viewer

- pyqtgraph

IDEAB

Animation

- Matplotlib has built-in animation
- http://matplotlib.org/examples/animation/index.html
- http://stackoverflow.com/questions/13316397/matplotlib-animation-no-moviewriters-available

IDEAB