Python in Web Development

Recap

- Vagrant
- Linux & Shell
- Git

LAB 2

Introduction to Python

AGENDA

- philosophy
- syntax
- primitives
- scoping
- core data structures
- essential modules/packages

Python 2 vs 3

- New features only in Python 3
- Python 2.7 will not be maintained past 2020
- Django 1.11 last version to support Python 2.7
- Highlights for Python 3:
  - Backwards incompatible
  - All strings are unicode by default
  - New syntax for **async**, **yield from**
  - Some built-in functions are lazy(range, filter)

Philosophy

- Zen of Python: try it!
  >>> import this
- PEP8
- Open and Friendly Community
- Who's using it:
  - Scientific Community: eg. NASA
  - Movie Industry: Walt Disney, Lucas Films
  - Internet Companies: youtube, google, dropbox
  - Machine Learning: Deepmind (TensorFlow)

Language Syntax

- Whitespace
- Keywords (eg: if, while, for, try, class, import)
- Context processors (**with**)
- Comprehensions (list, dict)
- Literals: (), [], {}, 42342L, 1101b
- Operator:
  - Arithmetic: +, -, *, **, %, /, //
  - Special: @
- Readability, easy to read and comprehend

Primitives

- True, False, None
- Int, float, complex
- Strings

Scoping

- Global scope
  - **global** keyword
- Class scope
  - Acts like a module
  - In methods you can use **self** (by convention)
- Function scope
  - You can access parent scope
  - Closures

Scoping - Example

- https://asciinema.org/a/bsmds690atuxjebt158tmvl1c

Core data types and structures

- **List: [1, 2, “a”]**
  - Mutable
  - Indexable
- **Tuple: (1, 2, “a”)**
  - Immutable
- **Set: {1, 2, “a”}**
  - Mutable (for immutable sets, use frozenset())
  - Unique values
- **Dictionary {“a”: 1, “b”:2, “c”: 3}**
  - Mutable
  - Unordered
  - Hashable keys

Essential modules / packages

- **Collections**
  - OrderedDict
  - DefaultDict
  - NamedTuple
- **Itertools**
  - count
  - chain
- **Functools**
  - wraps
  - partial
- Operator
- Json

Problems

Apply the concepts you learned

Resources

- Python control flow
- Python Data Structures
- Python Modules
- PEP Index
- Codewars katas
- When in doubt, go here

#1 Invert a given list of integer values

invert([1,2,3,4,5]) == [-1,-2,-3,-4,-5]
invert([1,-2,3,-4,5]) == [-1,2,-3,4,-5]
invert([]) == []

#2 Count all the occurrences of a word in a given string

http://bit.ly/2natiAv

string = “albatros are blue”
count(*string*) → albatros 1
are 1
blue 1

- For extra points, sort alphabetically

#3 Given a depth, a list of lists or integers, flatten it.

flatten([1, 2], 1)= [1, 2]
flatten([1, 2, [3]], 1) = [1, 2, 3]
flatten([1, 2, [3, [4]]], 2) = [1, 2, 3, 4]

#4 Given a list of integers, generate all the subsets

generate([1, 2, 3]) → [set(), {1}, {2}, {3}, {1, 2}, {1, 3}, {2, 3}, {1, 2, 3}]

Quiz time :)

https://goo.gl/forms/wdY8TuDnB9DCiVH72

Homework

- Learn Python the hard way, exercises 1 - 12 and 18
  - https://learnpythonthehardway.org/python3/

Thank you!