Python Basics

Functions

Markus Dale, 2018

Python - organizing code

Functions

- code reuse
- minimize repetition (DRY principle - don't repeat yourself)
- procedural programming

Types of functions

- Built-in functions (no import necessary): `len(str)`, `open`, `sorted`, `map`...
- Core module functions:
  - `import math, os`
  - `math.sqrt()`, `os.getcwd()`...
- Third party module functions:
  - Must be installed, e.g. `pipenv install requests`
  - `import requests`
  - `response = requests.get("http://docs.python.org")`
- User-defined - def, lambda

Named functions - learning/functions/defs.py

- a def function definition is an object with a name
- function is not executed until called at runtime

```
def function_name(arg1, arg2):
    statements
    return result


def foobar(arg1):
    if arg1 > 100:
        return 'foo'
    else:
        return 'bar'
```

Functions are first-class objects - learning/functions/first_class_objects.py

- def 'name' assigns 'name' as a reference to the function object
- Can assign new reference name
- Can pass functions to other functions as arguments (closure)
- Can return a function (closure) from a function

Recursive functions

- A function that calls itself with some ending condition

lambda - Anonymous Functions - learning/functions/lambdas.py

- lambda is an expression (def is a statement)
- can be used inside of a function call

```
words = ['this', 'is', 'a', 'test', 'of', 'the']
lengths = map(lambda w: len(w), words)

nums = [('apple', 1), ('tea', 4), ('beer', 2)]
most_frequents =
    sorted(nums,key=lambda t: t[1],reverse=True)
```