

# Programming Workshop: API and Project 1

Intermediate Workshop #5  
10 October 2016

# Outline

1. Speed Exercises
2. Quick Review:
  - a. Functions
  - b. Classes
  - c. When to Use Either
3. API - what is it, why make it, and how to make one
4. Project 1 (if time)

# Speed Exercises

# 1:

Make a function that accepts a list of strings and sorts them in alphabetical order

180 seconds

# Python - Classes

- Meant to allow for reusable, abstractable, extendable code
  - It's the core principle behind OO paradigm
- More complex
  - Less easy to read
  - Less easy to debug
  - Absolutely must have thorough documentation, thorough application testing, and thorough unit testing
  - Harder to make meanings or operations clear

# Python - Functions

- Meant to perform a generic operation on some specific input format
  - Data processing? Use a function.
  - Noticing a repeated operation? Use a function.
- Simple
  - Easy to read
  - Easy to debug
  - Should be documented, but not absolutely required

# Functions vs Classes

- Functions can be used to solve any issue. Classes have a specific purpose.
- Functions can call functions - but if a function has a subfunction, use a class.
- If a class has only one method, consider using a function.
- **If you do not have time to thoroughly build documentation, application tests, and unit tests for a class, *do not use classes.***

# API - What is it?

- “Application Programmer Interface” - API
- Provides user (client, server, human, other program, signal, etc.) with precise instruction set and precise set of possible responses (“call”s and “response”s)
- Is any set of user functions meant to trigger a known response
  - Many formal formats (REST, JSON, Swagger, etc.), but can also have informal formats

# API - What is it? (cont'd)

- API will *wildly* change depending on target
  - Hardware
  - Web service
  - Simple program
  - Etc.



# API - Why make it?

Allows for simplification, abstraction, speed, and accuracy.

Weather website example:

- Want to have a program read known weather service website and report to some program the expected weather for tomorrow
  - Can have program enter site, read site, interpret site, and report back with findings - like a human
  - Can use an API to send a request to the website host, host then sends back requested information in a known format

# API - Why make it? (cont'd)

Cell phone example:

- Want to provide GPS services for range of cell phone models
  - Each phone could directly interface with each type of GPS chip
  - GPS chip manufacturers could prescribe to a known API -

`my_phone_gps.where_am_i( ) = you_are_here`

See this Quora post for more examples and scenarios:

<https://www.quora.com/What-is-an-API-4>

# API - How to make one

1. Start with list of actions the user (or types of users) should take
  - Once list made, go over to reduce dependencies and over-complications
    - Ex: remove(), delete(), extract()
2. Make descriptive, brief names for each action (harder than it sounds!)
3. Write a document explaining *exactly* what the API call needs to receive in the call from the user and *exactly* what the API will return in the response. Include precise formats.
4. Write the code.

# Python - Class Example (cont'd)

```
1 class Kiln():
2     def __init__(self, input_temperature=78):
3         self._contents = []
4         self._temperature = input_temperature
5
6     def add_pottery(self, added_pottery):
7         for pottery in added_pottery:
8             self._contents.append(pottery)
9             if pottery.get_bake_temperature() <= self._temperature:
10                 pottery.mark_as_baked()
11
12     def change_temperature(self, new_temperature):
13         self._temperature = new_temperature
14         for pottery in self._contents:
15             if pottery.get_bake_temperature() <= self._temperature:
16                 pottery.mark_as_baked()
17
18     def get_temperature(self):
19         return self._temperature
20
```

# Python - Class Example (cont'd)

```
21     def get_contents(self, state=None):
22         if state:
23             pottery_of_specified_state = [x for x in self._contents if x.get_bake_status() == state]
24             return pottery_of_specified_state
25         return self._contents
26
27     def list_contents(self, state=None):
28         if state:
29             pottery_of_specified_state = [x for x in self._contents if x.get_bake_status() == state]
30             for pottery in pottery_of_specified_state:
31                 if pottery.get_description():
32                     print("%s: bake temperature=%i, description=%s" %(pottery.get_name(),
33                             pottery.get_bake_temperature(), pottery.get_description()))
34                 else:
35                     print("%s: bake temperature=%i" %(pottery.get_name(), pottery.get_bake_temperature()))
36         else:
37             for pottery in self._contents:
38                 if pottery.get_description():
39                     print("%s: bake temperature=%i, description=%s" %(pottery.get_name(),
40                             pottery.get_bake_temperature(), pottery.get_description()))
41                 else:
42                     print("%s: bake temperature=%i" %(pottery.get_name(), pottery.get_bake_temperature()))
```

# Python - API for Class Example

DOCUMENTATION: Kiln()

add\_pottery([list of pottery objects to add to kiln]):

Adds list of pottery objects to kiln

Returns nothing

change\_temperature(new temperature):

Changes temperature to new temperature

Iterates through objects in kiln to mark baked objects as baked

get\_temperature():

Returns temperature of kiln

get\_contents(state = desired state [optional]):

Given the optional state ("baked" or "unbaked"), returns list of items in the kiln marked with that state

Without optional state, returns list of all items in kiln

list\_contents(state = desired state):

Given the optional state ("baked" or "unbaked"), prints list of items in the kiln marked with that state

Without optional state, prints list of all items in kiln

Print is of the form:

Item\_name: bake\_temperature, description [if present]

Returns nothing

# Project 1 - Classes and APIs

In groups of 2 or 3:

- Using Python and a class, make a calculator
- Calculator has following methods:
  - Add
  - Subtract
  - Exponent
  - Clear
- Calculator has following attributes:
  - `current_value`
- Return `current_value` after each operation
- Make an API discussing use of your calculator

# Next Week:

- Finish Project 1
- How to approach programming problem