

2nd Half Slide Compilation

SI Session 10/17

By Cody Ptacek

Activity: Short Function Boardworks

Question 1:

Write a function for calculating the percentage of a provided part to a whole, and returns the answer as a string of the form: XX%

Question 2:

Write a function that converts fahrenheit to celsius:

$$c = 5/9 * (f - 32)$$

Question 2.5:

Write a function that converts celsius to fahrenheit.

Question 3:

Write a function that converts fahrenheit to kelvin. Use your function from question 2 as part of the calculation.

$$K = c + 273$$

Question 4

Write a function that prints the text for a menu.

Example Output:

=====Boardwork Menu=====

- 1) New Boardwork
- 2) Open Boardwork
- 3) Exit

Activity: Large Function

Task

- Write a function that takes in a string as a parameter. Within this string will be sentences ending in:
 - .
 - ?
 - !
- Have the function return a string with the same text, but with each sentence on a new line. (Make a new string with the `\n` character after each punctuation)

Input:

```
my_string = "To err is human. To akdkld is gibberish."  
print(sentence_seperator(my_string))
```

Output

To err is human.

To akdkld is gibberish

Activity: Session Review

Review Questions

- What is the purpose of a function?
- When do we want to use functions?
- What is the general template for a function?
 - Specify which elements are optional

SI Session 10/19

By Cody Ptacek

Activity Part 1: Short Function Questions

Question 1:

Write a function that returns the force of an object based on its mass and acceleration.

$$F = m * a$$

Question 1.1:

Write a function that returns the work of an object based on its mass, acceleration, and displacement. Use your force function as part of your work function.

$$W = F * d$$

Question 1.2:

Write a function that returns the power of an object based on its mass, acceleration, displacement, and time elapsed. Use your work function as part of the calculation.

$$P = w / t$$

Question 1.3:

Write a function that prompts the user for a mass, acceleration, displacement, and time. (all of which are float values). Return these answers in a list.

Activity Part 2: Combining Short Functions

Task

- In a main function, collect five sets of masses, accerations, displacements, and times from a user.
- Compare which instance had the highest displacement and which had the highest power. Print this data to the user.
 - It might be a good idea to create a new function for these comparisons, but you don't have to.

Example output of the comparisons:

Dataset 1 had the higher displacement of 10m

Dataset 2 had the higher power of 23 joules

SI Session 10/24

By Cody Ptacek

Activity: Diagramming Memory

Background

- Understanding how information is stored within memory for variables and lists is important for understanding how your code works.

Task

- Draw an area for a stack and a heap.
- Show how the code to the right would allocate data in the stack and the heap.
(What Gibbons has been showing in his notes)

```
def list_changer(user_list, user_num):  
    if len(user_list) >= 3:  
        user_list[2] = user_num  
    elif len(user_list) != 0:  
        user_list[0] = user_num  
  
def main():  
    my_list = [5, 10, 15, 20]  
    my_num = 77  
    list_changer(my_list, my_num)
```

Activity: List Editing Functions v List Returning Functions

Task 1

- Create a function that takes a list of words and capitalizes the first letter of each word in the list.
- NOTE: we are editing the list passed to us, NOT returning a new list.

Task 2

- Create a function that takes a list of words and RETURNS a new list that is the same except the first letter of each word in the new list is capitalized.
- DO NOT edit the original list!

HINT

- You are allowed to use the string method `capitalize()`.
 - `my_string = 'hello everyone!'`
 - `my_string.capitalize()`
 - Returns 'Hello everyone!'

Activity: Encryption Deciphering!

Task

- Create a function that takes a string and a dictionary as parameters.
 - The string will be a secret phrase that needs to be deciphered.
 - The dictionary contains a series of key-value pairs where the key is the character in the cypher and the value is the character in normal english.
- Your function should decipher the string, converting each character of the encrypted string to a character in regular English.
- Return the deciphered string.

Example String: 'xcoop'

Example Dict:

- `my_dict['x'] = 'h'`
- `my_dict['c'] = 'e'`
- `my_dict['o'] = 'l'`
- `my_dict['p'] = 'o'`

Deciphered string: 'hello'

SI Session 10/26

By Cody Ptacek

Activity: Dictionary Information

Task

- Show how to create an empty dictionary.
- Show how to create a dictionary that already contains a key-value pair.
- List the dictionary methods for getting...
 - Keys
 - Values
 - Key-Value Pairs

Activity: Reading in a Dictionary

Task 1

- Create a function that accepts a filename as a parameter.
 - Each line of the file will contain a word and its definition in the following manner:
 - word = definition
- Read these words and their definitions into a dictionary.
- Return the dictionary.

Task 2

- Create a function that searches a provided dictionary for a provided word.
- If the word is in the dictionary, return the definition of the word.
- If the word is not in the dictionary, ask the user if they would like to add the new word to the dictionary.
 - If yes, let them add the word to the dictionary.

Activity: Dictionary Questions

Questions

- What differentiates dictionaries from lists?
- What are the necessary features of the key of a dictionary?
- How do you add a new item from a dictionary?
- How do you remove an item from a dictionary?

SI Session 10/31

By Cody Ptacek

Activity: Modules

- What is our new keyword for including a module?
- What is the only module you have been shown so far?
- What functions does this module include?
 - You were shown five, try to name at least 3.

Activity: Functions with Modules

Task 1

- Create a function that takes a list of names as a parameter.
- Pick five names at random from the list.
- **WARNING:** make sure the list has at least five names. If it doesn't pick all names to win.

Task 2

- Create a random secret number between one and ten (inclusive).
- Prompt the user to pick a number between one and ten. (Inclusive)
- If the user picks the number, congratulate them for winning.
- Otherwise, tell them sorry that they lost.

SI Session 11/09

By Cody Ptacek

Activity: Sets and Parameter Review

Sets questions

- What is the unique quality of sets?
- What ways do we have to make new sets?
 - How do you make an empty set?
 - A set with initial values?
- What set methods have you seen thus far?

Parameter Questions

- What are the three 'types' of parameters?
- What special characters/order do we have to use to specify which parameters are which 'type'?

Activity: Set Functions

Task 1:

- Make a function that takes two sets as parameters.
 - Set 1 must be specified as positional only
 - Set2 must be specified as keyword only & contain a default value of the set of numbers 1-10
- If the two sets are the same, return the set.
- If the sets are different, return a new set consisting of all elements unique to each set. (Those only in set1 and only in set2).

Task 2

- Run through your code with the following calls and write the values that are returned for each call:
 - Just passing the set {4, 6, 8}
 - Passing the set {1, 2, 3, 4, 5, 6, 7, 8, 9}
 - Passing the sets {3, 6, 9} & {2, 4, 8}

SI Session 11/16

By Cody Ptacek

Activity: List Comprehension Review

List Comprehensions

- What is the purpose of a list comprehension?
- Why would you use a list comprehension over other methods of generating lists (i.e. functions)
- What is the general syntax of a list comprehension?

Activity: Simple List Comprehensions

Task 1:

- Create a list comprehension that returns a list of all of the numbers in `my_nums` that are not divisible by 7.

Task 2:

- Create a list comprehension that returns a list of all words in `my_words`, where each word has been stripped down and converted to lowercase.

Task 3

- Use a list comprehension to return a list of capitalized words with 3 exclamation points added to the ends from `my_words` if the original words were longer than 3 characters and words don't already contain punctuation.
 - [For this one, assume they only punctuation they might contain are '.', '!', or '?'. I don't want you to have to check for all types of punctuation!]

SI Session 11/28

By Cody Ptacek

Activity: Class Review

Class Review

- Classes are a huge topic in programming!
There will be a lot of new syntax thrown your way. Let's review what we've learned thus far...

- Where do we make a new class?
- How do we make a new class?
- How should the class be named?
- What is the one 'magic method' you all have learned thus far?
 - [It was `__init__()`. Wanted to clarify since you've learned more at this point]
 - Bonus: what does it do?
- Things do you include as parameters in the magic method?
 - Hint: for 168 purposes, you should ALWAYS have at least 1 Don't be like me and always forget this!
- How do you make member variables for classes?
- How do you define methods for classes?

Activity: Code Interpretation

```
my_recipe = {"butter":(1, 'cup'), "sugar":(1, 'cup'), "vanilla":(2, 'tsp')}  
my_pantry = {"butter":(0.5, 'cup'), "baking soda":(5, 'tbs'), "vanilla":(28, "tsp")}  
to_buy = dict()
```

```
for ingredient, quantity in my_recipe.items():  
    if ingredient in my_pantry.keys():  
        if quantity[0] > my_pantry[ingredient][0]:  
            to_buy[ingredient] = quantity  
    else:  
        to_buy[ingredient] = quantity
```

```
print(to_buy)
```

What will be printed to the console?

Activity: Mini Review Questions

How do you make an empty list?

How do you make an empty set?

How do you make an empty dict?

How do you add things to a list?

How do you add things to a dict?

How do you add things to a set?

Activity: Function Calls

Given the following function definition:

```
def the_ultimate_function(name, date_of_birth, /, star_sign, meyers_briggs_personality, *, fav_pokemon):
```

Which of the following function calls are valid? Which will throw errors?

- 1) `the_ultimate_function("Cody", "12/15/1995", "Sagitarious", "INFP", fav_pokemon="Piplup")`
- 2) `the_ultimate_function(fav_pokemon="Piplup", "Cody", "12/15/1995", "Sagitarious", "INFP")`
- 3) `the_ultimate_function("Cody", "12/15/1995", fav_pokemon="Piplup", star_sign="Sagitarious", meyers_briggs_personality="INFP")`
- 4) `the_ultimate_function("Cody", "12/15/1995", "Sagitarious", "INFP", "Piplup")`

SI Session 11/30

By Cody Ptacek

Activity: Magic Methods Review

- What are the magic methods we've learned so far?
 - Hint: There are 9 of them!
- What do each of them do?

Activity: Pokeball Class

We're going to make a class for different types of Pokeballs

Pokeballs have a name, a price, and an effectiveness rate, all of which should be passed as parameters upon creation.

However, the effectiveness rate and the price should be private and have validations associated with them.

The effectiveness rate should be 1 or more.

The price should be non-negative

The default effectiveness rate should be 1.

Methods

- Create a `set_price()` method that does the validation check (and passes if it fails)
- Create a `set_effectiveness()` method that does the validation check (and passes if it fails)
- Create getters for both the effectiveness and the price.

Activity: Pokeball Class (Magic Methods)

Define all comparison operators for the Pokeball class.

Comparisons should be between effectiveness rates.

[Bonus Challenge] After defining two of the comparison operators, it is possible to write all other operators in terms of just those two. If you want a bit of a challenge, try doing this.

- Hint: Start by defining `__eq__` and `__gt__` or `__lt__`

Define the magic methods `__str__` and `__repr__`.

`__str__` should produce the following output:

Pokeball Name: Great Ball
Price: 600 Pokemon Dollars
Effectiveness Rate: 1.5x

Activity: Code Interpretation for Our Pokeball Class

Given the following code in a main file:

```
pokeball1 = Pokeball("Greatball", 600, 1.5)
pokeball2 = Pokeball("Ultraball", 1200, 2.0)
my_list = [pokeball1, pokeball2]
```

What would be the outcomes for the following checks based upon how we've made the Pokeball class.

- 1) `pokeball1 > pokeball2`
- 2) `pokeball1 != pokeball2 or pokeball2 < pokeball1`
- 3) `print(pokeball2)`
- 4) `print(my_list)`
- 5) `pokeball1 > 5`

Activity: Mini Review Questions

Question 1

- Create a list comprehension that iterates through a list of tuples of the form:
`my_list = [('+', 8, 'USD'), ('-', 4, 'Euros')]...`
and keeps only the number value from the tuple squared. (You don't need to worry about the sign value)

Question 2

- List all the functions from random that we covered? How is each used?

Question 3

- Create a set that contains your group's top 5 TV shows.
- Collect five TV shows from a user and store them in a new set.
- Then, using set methods, tell the user which shows you both had in common and which shows you had in your list that they didn't have in their list.

Session Review

Class Review

- What do we call a variable specific to a class?
- What do we call a function specific to a class?
- What are all the magic methods we have learned thus far?
 - List all 9

Set Review

- What are all the set methods you learned in class? What does each do?