MOD 3 DAY 3 - CODE EXPLAINED - Python Deeper Div

Tuesday, September 26, 2023 8:07 PM

ACT

```
import os

import csv

# Open and read csv

    # Read the header row first (skip this part if there is no header)

    # Read through each row of data after the header

        # Convert row to float and compare to grams of fiber
```

BONUS

```
import os

import csv

    # @NOTE: This time, we do not use `next(csv_reader)` because there is no header for this file

    # Read through each row of data after the header

        # Convert row to float and compare to grams of fiber
```

ve

TIVITY 1

e

Convert row to float and compare to grams of fiber.

ACTIVITY 2

```
# Unlike lists, dictionaries store information in pairs

# ------------------------------------------------------------

# Create a dictionary to hold the actor's names.

# Create a dictionary using the built-in function.

# A dictionary of an actor.

# Add an actor to the dictionary with the key "name"

# and the value "Denzel Washington".

# Print the actors dictionary.

# Print only the actor.

# A list of actors

# Overwrite the value, "Denzel Washington", with the list of actors.

# Print the first actor

# ------------------------------------------------------------

# A dictionary can contain multiple pairs of information

# ------------------------------------------------------------

# A dictionary can contain multiple types of information

# A dictionary can even contain another dictionary
```

ACTIVITY 3

Dictionary full of info

Print out results are stored in the dictionary

ACTIVITY 4

fish = "halibut"
# Loop through each letter in the string

and push to an array

List comprehensions provide concise syntax for creating lists

We can manipulate each element as we go

List Comprehension for the above

We can also add conditional logic (if statements) to a list comprehension

List Comprehension with conditional

ACTIVITY 5

initialize list of names for user input

Note that the i is unused and could be replaced by an _

Use a list comprehension to create a list of lowercased names

Use a list comprehension to create a list of titlecased names from the

lowercased names. Reference:
https://www.tutorialspoint.com/python/string_title.htm

https://www.tutorialspoint.com/python/string_title.htm

Print the invitations

ACTIVITY 6

Basic Definition

    # code goes here

Simple Function with no parameters

you use parentheses to run the code in a function

Simple function with one parameter

Think of the parameter `message` as a variable

You assign the string "Hello, World!" when you call the function

This is like saying `message = "Hello, World!"`

Functions can have more than one parameter

Supply the arguments (values) when calling the function

@NOTE: Order is important when supplying arguments!

We can also specify default values for parameters

Make a quesadilla using the default topping

Make a quesadilla with a new topping

Functions can return a value

You can save the value that is returned

You can also just print the return value of a function

ACTIVITY 7

Write a function that returns the arithmetic average for a list of numbers

Test your function with the following:

ACTIVITY 8

```
import os

import csv

# Path to collect data from the Resources folder

# Define the function and have it accept the 'state_data' as its sole parameter

    # For readability, it can help to assign your values to variables with descriptive
names

    # CSV headers: State or jurisdiction, Adjusted cohort (Public), Completers (Public),

    # Adjusted cohort (Nonprofit Private), Completers (Nonprofit Private),

    # Adjusted cohort (For-profit Private), Completers (For-profit Private)

    # Total students can be found by adding students of each category together

    # Total graduates can be found by adding graduates of each category together

    # Public grad rate can be found by dividing the total public graduates by the total
public

    # students and multiplying by 100
```

```
    # Note that some states do not have nonprofit or forprofit private schools, so data must be checked

    # for zeros

    # Nonprofit grad rate can be found by dividing the total nonprofit graduates by the total nonprofit

    # students and multiplying by 100

    # Forprofit grad rate can be found by dividing the total forprofit graduates by the total forprofit

    # students and multiplying by 100

    # Calculate the overall graduation rate

    # If the overall graduation rate is over 50, message is "Graduation success".

    # Otherwise it is "State needs improvement".

    # Print out the state's name and their graduation rates

# Read in the CSV file

    # Split the data on commas

    # Prompt the user for what state they would like to search for

    # Loop through the data

        # If the state's name in a row is equal to that which the user input, run the 'print_percentages()' function
```