MOD 3 Day 2 - EXPLAINED CODE - Reading, Writing, and Pyrithmetic

Tuesday, September 26, 2023 7:55 PM

ACTIVITY 1

```
# Print Hello User!

#print = ("Hello User")

# Take in User Input

# Respond Back with User Input

# Take in the User Favorite Number

# Respond Back with a statement based on your favorite number
```

ACTIVITY 2

```
# A For loop moves through a given range of numbers

# If only one number is provided it will loop from 0 to that number

# If two numbers are provided then a For loop will loop from the first number up until it reaches the second number

# If a list is provided, then the For loop will loop through each element within the list

# A While Loop will continue to loop through the code contained within it until some condition is met
```

ACTIVITY 3

```
# The list of candies to print to the screen

    "Starbursts",
# The amount of candy the user will be allowed to choose

# The list used to store all of the candies selected inside of

# Print all of the candies to the screen and their index in brackets

# Another option to run the for loop involves Python's enumerate method

# This method obtains both the index and the value of an item during a for loop

# for index, candy in candy_list:

#     print(index, candy)

# Run through a loop which allows the user to choose which candies to take home with them

    # Add the candy at the index chosen to the candy_cart list

# Loop through the candy_cart to say what candies were brought home
```

BONUS

```
# The list of candies to print to the screen

# The amount of candy the user will be allowed to choose

# The list used to store all of the candies selected inside of

# Print all of the candies to the screen and their index in brackets

# Set answer to "yes" for while loop
```

```
    # Ask which candy the user would like to bring ho

    # Add the candy at the index chosen to the candy_cart list

    # ask the user if they want more candy

# Loop through the candy_cart to say what candies were brought home
```

ACTIVITY 4

```
# Initial variable to track shopping status

# List to track pie purchases


# Pie List

# Display initial message

# While we are still shopping...

    # Show pie selection prompt

    # Add pie to the pie list

    # Inform the customer of the pie purchase

    # Provide exit option

# Once the pie list is complete
```

BONUS

```
# Initial variable to track shopping status

# List to track pie purchases

# Pie List

# Display initial message

# While we are still shopping...

    # Show pie selection prompt


    # Get index of the pie from the selected number

    # Add pie to the pie list by finding the matching index and adding one to its value

    # Inform the customer of the pie purchase

    # Provide exit option

# Once the pie list is complete

# Count instances of each pie

# Loop through the full pie list

    # Gather the count of each pie in the pie list and print them alongside the pies
```

ACTIVITY 5

```
# Store the file path associated with the file (note the backslash may be OS specific)

# Open the file in "read" mode ('r') and store the contents in the variable "text"

    # This stores a reference to a file stream
```

```
    # Store all of the text inside a variable called "lines"

    # Print the contents of the text file
```

ACTIVITY 6

```
# Import the random and string Module

# Utilize the string module's custom method: ".ascii_letters"

# Utilize the random module's custom method randint

for x in range(10):
```

ACTIVITY 7

JUST A READ ME

ACTIVITY 8

```
# Module for reading CSV files

# # Method 1: Plain Reading of CSV files

# with open(csvpath, 'r') as file_handler:

#     lines = file_handler.read()

#     print(lines)

#     print(type(lines))

# Method 2: Improved Reading using CSV module

    # CSV reader specifies delimiter and variable that holds contents
```

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

    csv_header = next(csvreader)
    # Read each row of data after the header
```

ACTIVITY 9

```
# Modules

# Prompt user for title lookup

# Set path for file

# Set variable to check if we found the video

# Open the CSV using the UTF-8 encoding

    # Loop through looking for the video

        # Set variable to confirm we have found the video

    # If the book is never found, alert the user
```

ACTIVITY 10

```
# Dependencies

# Specify the file to write to

# Open the file using "write" mode. Specify the variable to hold the contents

    # Initialize csv.writer

    # Write the first row (column headers)
```

```
    # Write the second row
```

ACTIVITY 11

```
import csv

import os

# Three Lists

# Zip all three lists together into tuples

# Print the contents of each row

# save the output file path

# open the output file, create a header row, and then write the zipped object to the csv

# # to print out to terminal:

# #comment out above code and run the code below

# for employee in roster:

#     print(employee)
```

ACTIVITY 12

```
import os

import csv

# Lists to store data

# with open(udemy_csv, encoding='utf-8') as csvfile:
```

```
# with open(duchy_csv, encoding='utf-8') as csvfile:

        # Add place

        place.append(row[0])
        # Add population

        # Add per capita income

        # Add poverty count

        # Determine poverty rate to 2 decimal places, convert to string

        # Split the place into county and state

# Zip lists together

# Set variable for output file

#  Open the output file

    # Write the header row

    # Write in zipped rows
```

ACTIVITY 13

```
# Define the function and tell it to print "Hello!" when called

# Call the function within the application to ensure the code is run

# Functions that take in and use parameters can also be defined

# When calling a function with a parameter, a parameter must be passed into the function

# ------------#
# The prime use case for functions is in being able to run the same code for different
```

values