Week 1 materials + answers

topics covered

- intro to programming: problem solving, different programming languages, statements, machine code
- python 3
- print(output) statement

some recap

- string is a series of characters
- to print a string use: `print("Your sentence")` or `print('Your sentence')`
- remember that everything is printed the way we read: top to bottom, left to right
- you can use as many print commands as you want
- Concatenation is the process of combining or joining two or more strings together to create a new string. The plus operator (+) is used for string concatenation in Python. `print("Hello, " + "everyone")` will give an output:

    > Hello, everyone

- When you put '\n' in a string, the print() function starts a newline, e.g. `print('Hello \n Bye')` will give an output:

    > Hello
    > Bye

- all files have some extension: .txt .jpeg .pdf   python files have extension .py

exercises

1. What is the difference between these two statements?

    a. print("My name is Bob")

    b. print('My name is Bob')

    answer: the output they produce is the same, meaning these two statements are interchangeable

2. What is the difference between these two statements?

    a. print("I LOVE DOGS")

    b. print("i love dogs")

    answer: first print statement will output a sentence with all uppercase letters, while second print statement will output a sentence with all lowercase letters. In Python, string literals are case sensitive, meaning that uppercase and lowercase letters are considered distinct characters. So, these two statements will produce different output when executed.

3. What is the output of the following print statement?

    ```
    print("Have a great day!")
    ```

    a. "Have a great day!"

    b. "Have a great day"

    c. Have a great day!

    d. 'Have a great day!'

    answer: c, explanation: when we want to print something we can use double or single quotation marks, however the output will be just the text between those marks(so no marks will be shown in the output)

4. What is the output of the following statements?

    ```
    print("Hi there!")
    print("How are you doing?")
    ```

    a. Hi there! How are you doing?

    b. How are you doing? Hi there!

    c. "Hi there!"

       "How are you doing?"

    d. Hi there!

       How are you doing?

e. "Hi there. How are you doing?"

answer: d, explanation: each `print()` statement outputs the provided text as a new line, there are no quotation marks in the output in console (look previous question)

5. Write a program that prints a message saying

I love Python!

answer: `print("I love Python!")` or `print('I love Python!')`

6. Write a program that prints a message saying

Hello, World!!!

answer: `print("Hello, World!")` or `print('Hello, World!')`

7. Write a program that prints a message saying your name and your age, e.g.

My name is Colin;) I am 20 years old!

answer: `print("My name is Colin;) I am 20 years old!")` or `print('My name is Colin;) I am 20 years old!')`

8. What would be the output of this statement?

```
print("Hello" + "World")
```

a. HelloWorld

b. Hello World

c. Hello

   World

d. Hello + World

answer: a, explanation: The plus operator (+) is used for string concatenation in Python. Concatenation is the process of combining or joining two or more strings together to create a new string. In this case, the strings "Hello" and "World" are concatenated together, resulting in the combined string `HelloWorld`. The output does not include any spaces or other characters between the two words.

9. What would be the output of this statement?

```
print("I " + "love " + "listening " + "to " + "music" + "!")
```

a. Ilovelisteningtomusic

b. I love listening to music!

c. I

   love

   listening

   to

   music

   !

d. I + love + listening + to + music + !

answer: b, The plus operator (+) is used to concatenate the given strings together. In this case, the strings "I ", "love ", "listening ", "to ", "music", and "!" are concatenated in the given order. The resulting string is "I love listening to music!".

10. Write this sentence using string concatenation:

    I like chocolate and vanilla ice cream.

    answer: `print("I " + "like " + "chocolate " + "and " + "vanilla " + "ice " + "cream.")`

11. Write this sentence using string concatenation:

    Today is a great day!

    answer: `print("Today " + "is " + "a " + "great " + "day!")`

12. Write a program to display the message "Welcome to Python" three times, on separate lines using three `print()` statements.

    answer:

    ```
    print("Welcome to Python")
    print("Welcome to Python")
    print("Welcome to Python")
    ```

13. Write a program to display the message "I am learning programming" five times, on separate lines using five `print()` statements.

    answer:

    ```
    print("I am learning programming")
    print("I am learning programming")
    print("I am learning programming")
    print("I am learning programming")
    print("I am learning programming")
    ```

14. Write a program to display the message "Python is awesome!" two times, on separate lines, using only one `print()` statement and the `\n` escape sequence, look for more info in the recap section.

    answer: `print("Python is awesome!\nPython is awesome!")`

15. Write a program that prints out your favorite food, followed by a blank line, followed by your favorite color, followed by a blank line, followed by your favorite animal. Use either `\n` escape sequence or separate `print()` statements. Save this program as p1_15.py.

    answer: this is just an example, your answer may be different

    ```
    print("Favorite food: Pizza")
    print()
    print("Favorite color: Blue")
    print()
    print("Favorite animal: Dog")
    ```

16. Write a program that prints out your favorite hobby, followed by a blank line, followed by your favorite sport, followed by a blank line, followed by your favorite game. Use either `\n` escape sequence or separate `print()` statements. Save this program as p1_16.py.

    answer: this is just an example, your answer may be different

    `print("Favorite hobby: Playing guitar\n\nFavorite sport: Tennis\n\nFavorite game: Chess")`

17. Print a pattern of your choice using asterisks (*). For example, you can print a triangle, a square, or any other shape you like.

    answer: this is just an example, your answer may be different, here is a triangle printed using asterisks

    ```
    print("   *")
    print("  ***")
    print(" *****")
    print("*******")
    ```

18. Print some of the lyrics of your favorite song.

    answer: this is just an example, your answer may be different, this is a song called "Dance again" by Selena Gomez

    ```
    print("I kick-start the rhythm")
    print("All the trauma's in remission")
    print("No, I don't need permission")
    print("Feels so, feels so, feels so good to dance again")
    ```

19. Write a program that prints the to-do list for each day of the week. You should include a separate `print()` statement for each day of the week and use `\n` within that day to print the list, each day's to-do list should be displayed on a new line. Make sure to include a descriptive label for each day.

Here is an example of how you might display your to-do list:

Wednesday:

- Study for the exam

- Clean the house

- Take a walk in the park

answer: this is just an example, your answer may be different

```
print("Monday:\n- Finish homework\n- Go to the gym\n- Call mom\n")
print("Tuesday:\n- Attend a team meeting\n- Prepare presentation\n- Buy groceries\n")
print("Wednesday:\n- Study for the exam\n- Clean the house\n- Take a walk in the park\n")
print("Thursday:\n- Work on a coding project\n- Attend a language class\n- Have dinner with friends\n")
print("Friday:\n- Plan weekend activities\n- Watch a movie\n- Relax and unwind\n")
print("Saturday:\n- Go hiking\n- Read a book\n- Cook a special meal\n")
print("Sunday:\n- Spend time with family\n- Plan for the upcoming week\n- Enjoy a day of rest\n")
```

EXTRA TASK:

In Python, you can use `upper()` function. It is a built-in string method that converts all the characters in a string to uppercase. It returns a new string with the uppercase representation of the original string. To use a function simply use a '.' after the string you want to convert to uppercase. For example, `print("hello".upper())` will output: HELLO. The `upper()` function is just one example of the many string methods available in Python.

So your task is to write a program that prints out a motivational quote in the following format:

DREAM BIG
and
ACHIEVE GREATNESS!

You can use your favorite quote or find one on the internet, or simply use the one above. You need to use `.upper()` function, concatenation and `\n` escape sequence in this exercise.

answer: `print("dream big\n".upper() + "and\n" + "achieve greatness!".upper())`