Week 2 materials

topics covered

- comments
- variables and assignment
- naming conventions
- data types
- user input
- beep

some recap

comments:

- to create a comment in Python use `#` at the beginning of the line, e.g.

```
# This is a comment in Python
# It is used to provide additional information or explanations in the code
# Comments are ignored by the interpreter and do not affect the program's execution
```

variables:

- you can view a variable as a box with a label on it
- to create a variable, you just assign it a value and then start using it, e.g. `price = 100` assigns price variable to 100
- you can change/substitute the value of the variable (reassign the value), e.g. `price = 300` within the same program will reassign price variable to 300
- variable names in Python can be any length and can consist of uppercase and lowercase letters (A-Z, a-z), digits (0-9), and the underscore character (_); variable names are case sensitive
- however, the first character of a variable name cannot be a digit: `1099_apple` is NOT a valid variable name

common naming conventions(when you need to use several words to name one variable):

Snake case (e.g: some_variable) — mostly used in Python

Camel case (e.g: someVariable)

Pascal case (e.g: SomeVariable)

reserved words in python(cannot name a variable like this):

`False`	`def`	`if`	`raise`
`None`	`del`	`import`	`return`
`True`	`elif`	`in`	`try`
`and`	`else`	`is`	`while`
`as`	`except`	`lambda`	`with`
`assert`	`finally`	`nonlocal`	`yield`
`break`	`for`	`not`	
`class`	`from`	`or`	
`continue`	`global`	`pass`	

data types:

`string` — surrounded by “” or ‘’ (e.g. ‘Tom’ or “I am 30 years old” or “1000”)

`integer` — whole number (e.g. 4 or 35 or 1000)

`float` — number with a decimal point (e.g. 4.7 or 36.154 or 1000.125)

`boolean` — can be either True or False (e.g. is_working = False or is_playing_violin = True)

note: True & False should start with uppercase letter

printing variables:

- to print the variable do NOT use “” or ‘’ in print() statement, simply write the name of the variable inside print(), e.g. `print(price)`
- to print multiple variables without additional text simply pass the variables as arguments to the `print()` function, separating them with commas. For example: `print(variable1, variable2, variable3)`. In this case, the values will be printed, separated by spaces
- in Python, the `str()` function is used to convert a value into its string representation. It takes an argument, which can be of any data type, and returns a string representation of that value
- if you want to include not only the variable, but also some text, here are several methods how:

  - concatenation operator (+): Use the concatenation operator (+) to combine text and variables. If variables are numbers use `str()` For example: `print("The price is: " + str(price))`

  - f-strings (formatted string literals): Starting from Python 3.6, you can use f-strings, which provide a concise way to include variables inside a string. Simply prefix the string with `f` and use curly braces `{}` to enclose the variable. For example: `print(f"The price is: {price}")`

escape character:

In programming, an escape character is a special character that is used to give a different meaning to another character or sequence of characters.

Think of the escape character as a secret code that you can put in front of certain characters to tell the computer to handle them differently. It helps you include special

characters or create specific behaviors within your code.

The most common escape character is the backslash (`\`). When the backslash is placed before a character or sequence of characters, it changes their interpretation. For example, if you want to include a double quote (`"`) within a string that is enclosed by double quotes, you can use the escape character before the double quote to indicate that it should be treated as a literal character, rather than the closing delimiter of the string. Same with `'`.

Another way is use double quotes for that string where you want to include `'`

Use single quotes in a string where you want to include `"`

print(”I’m happy”) — correct

print(’I\’m happy’) — correct

print('I said, "Hello!"') — correct

print("I said, \"Hello!\"") — correct

print(’I’m happy’) — incorrect

print("I said, "Hello!"") — incorrect

explain why the last two statements are incorrect?

user input:

- to receive the input from the user use function called `input()`
- inside parentheses you can write a prompt for the user: `input(’What is your name? ’)`
- we can store the result from the user’s input in a variable: `name = input(’What is your name? ’)`
- then we can use `name` just as a regular variable!
- `input()` function always returns data as a string
- we can convert result into an integer, or float by calling the built-in `int()`, `float()` functions, e.g. `price = float(input('Enter the price: '))`

exercises

1. What will happen after execution of this program?

```
#print("I like reading")
#print('I don\'t like olives')
```

2. Create a variable called `name` and assign your name to it. Then, print the value of the variable.

3. Create a variable called `fruit` and assign a string representing your favorite fruit to it. Then, print a message using the variable like this: "I love <fruit>!". Use concatenation (for more info look week 1)

5. Create a variable called `greeting` and assign a string representing a greeting of your choice (e.g., "Hello", "Hi", "Hey"). Then, create another variable called `name` and assign your friend's name to it. Print a personalized greeting like this: "<greeting>, <name>! How are you today?"

5. variable names: valid or invalid?
    a. Is the variable name `TotalMarks` correct?
    b. Is the variable name `number-of-students` correct?
    c. Is the variable name `firstName` correct?
    d. Is the variable name `myVar1` correct?
    e. Is the variable name `customerName` correct?
    f. Is the variable name `productPrice` correct?
    g. Is the variable name `3rdStudent` correct?
    h. Is the variable name `isAvailable?` correct?
    i. Is the variable name `total-sales` correct?
    j. Is the variable name `customer_email` correct?

6. What is the data type of each variable?
    a. What is the data type of the variable 'age'? `age = 25`
    b. What is the data type of the variable 'name'? `name = "John Doe”`
    c. What is the data type of the variable 'price'? `price = 9.99`
    d. What is the data type of the variable 'is_valid'? `is_valid = True`
    e. What is the data type of the variable 'quantity'? `quantity = 10`
    f. What is the data type of the variable 'message'? `message = 'Hello'`
    g. What is the data type of the variable 'discount'? `discount = 0.2`

7. Look at this sentence: "The temperature is 25.5 degrees Celsius."
    a. Create a variable to represent the temperature and set it as it is in the sentence.
    b. Print the above sentence using the variable. Use one of the methods from some recap section

    Use meaningful name for the variable and correct naming convention, use comments where needed

8. Look at this sentence: "There are 10 students in the class."
    a. Create a variable to represent the number of students and set it as it is in the sentence.
    b. Print the above sentence using the variable.

    Use meaningful name for the variable and correct naming convention, use comments where needed

9. Look at this sentence: "The event will take place on November 15, 2022, from 9:00 AM to 5:00 PM."
    a. Create a variable to represent the event date and set it as "November 15, 2022".
    b. Create a variable to represent the event start time and set it as "9:00 AM".

c. Create a variable to represent the event end time and set it as "5:00 PM".

d. Print the above sentence using the variables.

Use meaningful names for the variables and correct naming conventions, use comments where needed

10. Look at this sentence: "The price of the book is $19.99, and it has 350 pages."

    a. Create variables to represent the following data: the price of the book, the number of pages in the book, set them as they are in the sentence above.

    b. Print the above sentence using the variables.

    c. Reassign the price to 25.99 and number of pages to 450.

    d. Print the final sentence using the updated variables.print the final sentence.

11. Look at this sentence: "The duration of the movie is 2 hours and 30 minutes, and it has a rating of 8.5 out of 10."

    a. Create variables to represent the duration of the movie and its rating, and set them as they are in the sentence above.

    b. Print the above sentence using the variables.

    c. Reassign the duration to 2 hours and 45 minutes, and the rating to 9.0 out of 10.

    d. Print the final sentence using the updated variables.

12. Write a program that asks the user for their name using the input() function. Store the name in a variable and print a personalized greeting using the variable.

13. Write a program that prompts the user to enter their favorite color and favorite animal using the input() function. Store these values in separate variables and concatenate them to create a sentence of your choice. Print the sentence using the variables and additional text.

14. Write a program that prompts the user to enter their favorite number using the input() function. Store the number in a variable and print a message that includes the user's favorite number.

15. Write a program that asks the user for various words (e.g., noun, verb, adjective) using the input() function. Store each word in a separate variable. Then, use the variables to create a funny sentence by filling in the blanks. Finally, print the completed sentence.

    You can think of your own sentence, or use any of these sentences

    - I saw a **noun verb** down the street wearing a **adjective** hat.
    - The **adjective noun** decided to **verb** through the grocery store in a tutu.
    - Yesterday, I found a **noun** trying to **verb** in the library while wearing a **adjective** wig.
    - At the zoo, I witnessed a **adjective noun** trying to **verb** across the exhibit while balancing a **adjective** umbrella.

    Example output:

    ```
    Enter a noun: cat
    Enter a verb ending with -ing: sneezing
    Enter an adjective: gigantic

    I saw a cat sneezing down the street wearing a gigantic hat.
    ```

16. Write a program that asks the user for a date of birth, create appropriate variables, convert them into integers, print the date in a format: Day: 2, Month: 4, Year: 2000. Save this program as p2_16.py

17. Write a program that asks the user to enter their dream travel destination using the `input()` function. Store the input in a variable and assign it a suitable name. Store the message of your choice in another variable and concatenate it with dream travel destination variable. Then, print a message. Save this program as p2_17.py

18. Create and run the beep program below

    The `\a` is an escape sequence that represents the alert or bell character. When this character is encountered in a string, it produces a sound or visual alert, depending on the system or environment where the code is executed.

    By printing `\a \a \a`, the program will make a beep sound 3 times

    ```
    # beep.py: beeps 3 times!!!

    print("\a \a \a")
    ```

    Add one beep after each print statement to exercise 12 of the first week.

    Note: you can use `time.sleep(1)` after each beep, but add `import time` to the very beginning of your program. The `time.sleep(1)` function is used to introduce a 1-second delay after each beep sound.

19. Create a program that simulates a hospital registration system. Prompt the user to enter the following information:

    - Name
    - Surname
    - Age
    - Height (in cm)
    - Health complaints

    Store each piece of information in a separate variable with an appropriate name.

    Finally, print the information in the following format:

    Name: [Name]
    Surname: [Surname]
    Age: [Age]
    Height: [Height]
    Health complaints: [Health Complaints]

    Replace [Name], [Surname], [Age], [Height], and [Health Complaints] with the actual values entered by the user.

Save program as p2_19.py

Example output:

```
Enter your name: Rory
Enter your surname: Gilmore
Enter your age: 21
Enter your height (in cm): 168
Enter your health complaints: Pain in the back

Name: Rory
Surname: Gilmore
Age: 21
Height: 168
Health complaints: Pain in the back
```

EXTRA TASK:

Create a program that acts as a personal information tracker. Prompt the user to enter the following details:

1. Name: (string)
2. Age: (integer)
3. Height (in meters): (float)
4. Is Student: (string)
5. Hobbies (separated by commas, just 3 hobbies): (string)

Store each piece of information in a separate variable with an appropriate name. Ensure that you use the correct data type for each variable. Ask only for 3 hobbies!!!

Next, use the variables to print out a summary of the entered information. The output should follow this format:

This is your personal Information:
Name: [Name]
Age: [Age]
Height: [Height] meters
Is Student: [Is Student]
Hobbies: [Hobby1, Hobby2, Hobby3]

Replace [Name], [Age], [Height], [is_student], and [Hobby1, Hobby2, Hobby3] with the actual values entered by the user.

Use ONLY ONE print() statement!!!

Save program as p2_extra.py