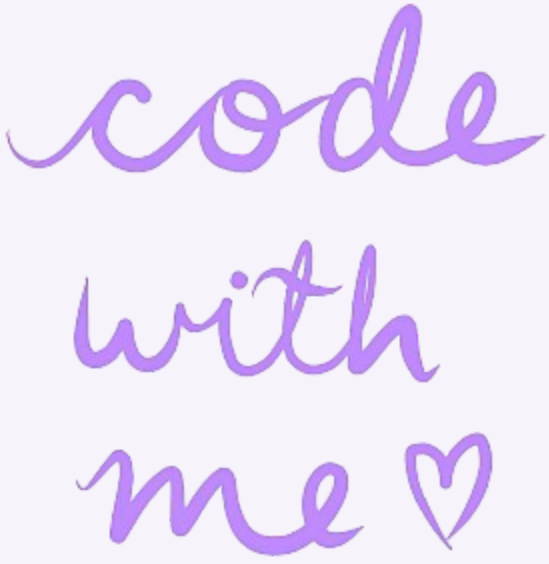
Code with Me Meeting #6



String Methods

STRING METHODS:

- String methods can do so many things to strings!
- You can call a string method on any string. There are over 30 string methods!
- We will be covering just a few of them.

UPPER AND LOWER:

- These methods allow you to change the entire string to lowercase, or the entire string to uppercase.
- To change the entire string to uppercase, `string.upper()` will return the entire string in CAPITAL LETTERS
- It's the same for lowercase: `string.lower()`
- You can also save these strings in another variable if you want to use it later.
- These are going to be super useful while doing If... Else Statements and Loops

LEN:

- Len is a function that allows you to find the length, in characters, of a certain string.
- Len isn't a true string method, but it works almost the same.
- You can write `len(string)` and it will return the number of characters in that string.
- You can also write `len("Hello World")` i.e, the actual string, not the variable
    - This will also return the length, in this case, 11
- Remember that spaces are also counted as characters.

COUNT:

- `count()` tells you how many times a particular character, phrase, etc. occurs in a string.
- `mystring.count("is")` will return the number of times "is" occurs in the string stored in the variable `mystring`
- `mystring.count(x)` can be used. Here, `x` is also a variable, and it can store a substring.
- You can also specify the start and end, which tells you what character you want to start counting from, and what character you want to end at.
- `mystring.count("is", 9, 23)`

FIND:

- `find("is")` finds the index (character count) of where the first occurrence of the substring is.
- You can also specify the start and end characters to start searching.
- Refer to examples
- You can use `rfind()` to start searching from the end of the string

REPLACE:

- `replace()` allows you to find a character, word, or phrase, and replace it with another.
- You can write `string.replace('o', 'e')`. This will replace all the 'o's in a string with 'e'.
- You can also specify the number of times you want to replace it.
- Ex: If you only want to replace the first 'o' with an 'e', write `string.replace('o', 'e', 1)`

SPLIT:

- `split()` splits the string into separate strings every time there's a space or `/n` unless you specify a different character to split at
- It puts the new strings into something called a list.
- You can then access these strings. We will learn more about this later in loops

JOIN:

- `join()` can actually join back a string that you have split!
- You can apply this to a list.
- If you have `mylist = [“My”, “name”, “is”, “Mickey”]`
- You can store a separator in `separator = “,”`
- Use `separator.join(mylist)`
- Make sure you store it in a variable so you can print it out later!
- The separator tells Python to join it using the separator you have stored in the variable. I used a comma.
    - It will return “My, name, is, Mickey”.