Day 3

Strings in Python

1. Known as `str` in python.
2. should start and end with the same code, eg if single then should end with single quote and so on.
3. To use multiple lines for a single string, you can use `"""` or `'''` , they are also known as strong type of strings.
4. They can be indexed, that means the letters can be accessed like list accessing.
5. ASCII vs UNICODE (8bit vs 16bit)
6. `ord()` *is used to convert char to ASCII, while* `chr()` *converts ASCII to character.*
7. If you multiply a string with a `int` python will print the string that `int` times.
8. OPERATORS:
    a. `+` : String Concatenation
    b. `*` :
    c. `[ ]` :

    1. `[ : ]` : String slicing
    2. `in` : checks if the sub string is contained in the string
    3. `not in` : checks if the sub string is not contained in the string
9. METHODS:
    a. `capitalize` : Changes the first letter into a capital letter
    b. `title` : Changes first char of all words into capital
    c. `uppercase` : converts all into uppercase
    d. `lowercase` : converts all into lowercase

e. `isUpper` : returns if the string is upper case

f. `isLower` : returns if the string is lower case

g. `count`: count how many times str occured in a string

h. `find` : returns the index else -1.

i. `index`: same as find but raises exception

j. `len` : returns the length of the string.

k. `rfind`: reverse find method

l. `rindex`: reverse index method

m. `center(width, fillchar)`: used in pattern printing, fills char around the string

n. `ljust(width, fillchar):` used in pattern printing, fills char right to the string

o. `rjust(width, fillchar)`: used in pattern printing, fills char left to the string

p. `replace` : replaces the word with given word.

q. `split`: when a string is given it breaks the string and converts it into a list of string

Functions

1. Re-usability
2. Support DRY principle.
3. faster, modular codding

Doc-strings (Document strings)

> it helps user to know about the function