1. Create a function that takes three integer arguments (a, b, c) and returns the amount of integers which are of equal value.

Examples:
equal(3, 4, 3) ➞ 2
equal(1, 1, 1) ➞ 3
equal(3, 4, 1) ➞ 0

Notes:
Your function must return 0, 2 or 3.
Ans:

In [40]:
```
def equal(a,b,c):
    output = 0
    if a == b == c:
        output = 3
    elif a == b or b == c or a == c:
        output = 2
    else:
        output = 0
    print(f'equal({a},{b},{c}) ➞ {output}')
equal(3, 4, 3)
equal(1, 1, 1)
equal(3, 4, 1)
```

```
equal(3,4,3) ➞ 2
equal(1,1,1) ➞ 3
equal(3,4,1) ➞ 0
```

2. Write a function that converts a dictionary into a list of keys-values tuples.

Examples:
dict_to_list({
"D": 1,
"B": 2,
"C": 3
}) ➞ [("B", 2), ("C", 3), ("D", 1)]
dict_to_list({
"likes": 2,
"dislikes": 3,
"followers": 10
}) ➞ [("dislikes", 3), ("followers", 10), ("likes", 2)]

Notes:
Return the elements in the list in alphabetical order.
Ans:

In [47]:
```
def dict_to_list(dict1):
    list1 = list(dict1.items())
    print(f'dict_to_list({dict1}) ➞ [{sorted(list1)}')
dict_to_list({"D": 1,"B": 2,"C": 3})
dict_to_list({"likes": 2,"dislikes": 3,"followers": 10})
```

```
dict_to_list({'D': 1, 'B': 2, 'C': 3}) ➞ [[('B', 2), ('C', 3), ('D', 1)]
dict_to_list({'likes': 2, 'dislikes': 3, 'followers': 10}) ➞ [[('dislikes', 3), ('followers', 10), ('likes', 2)]
```

3. Write a function that creates a dictionary with each (key, value) pair being the (lower case, upper case) versions of a letter, respectively.

Examples:
mapping(["p", "s"]) ➞ { "p": "P", "s": "S" }
mapping(["a", "b", "c"]) ➞ { "a": "A", "b": "B", "c": "C" }
mapping(["a", "v", "y", "z"]) ➞ { "a": "A", "v": "V", "y": "Y", "z": "Z" }

Notes:
All of the letters in the input list will always be lowercase.
Ans:

In [59]:
```
def mapping(list1):
    dict1 = {}
    for i in list1:
        dict1[i] = i.upper()
    print(f'mapping({list1})➞{dict1}')
mapping(["p", "s"])
mapping(["a", "b", "c"])
mapping(["a", "v", "y", "z"])
```

```
mapping(['p', 's'])➞{'p': 'P', 's': 'S'}
mapping(['a', 'b', 'c'])➞{'a': 'A', 'b': 'B', 'c': 'C'}
mapping(['a', 'v', 'y', 'z'])➞{'a': 'A', 'v': 'V', 'y': 'Y', 'z': 'Z'}
```

4. Write a function, that replaces all vowels in a string with a specified vowel.

Examples:
vow_replace("apples and bananas", "u") ➞ "upplus und bununus"
vow_replace("cheese casserole", "o") ➞ "chooso cossorolo"
vow_replace("stuffed jalapeno poppers", "e") ➞ "steffed jelepene peppers"

Notes:
All words will be lowercase. Y is not considered a vowel.

Ans:

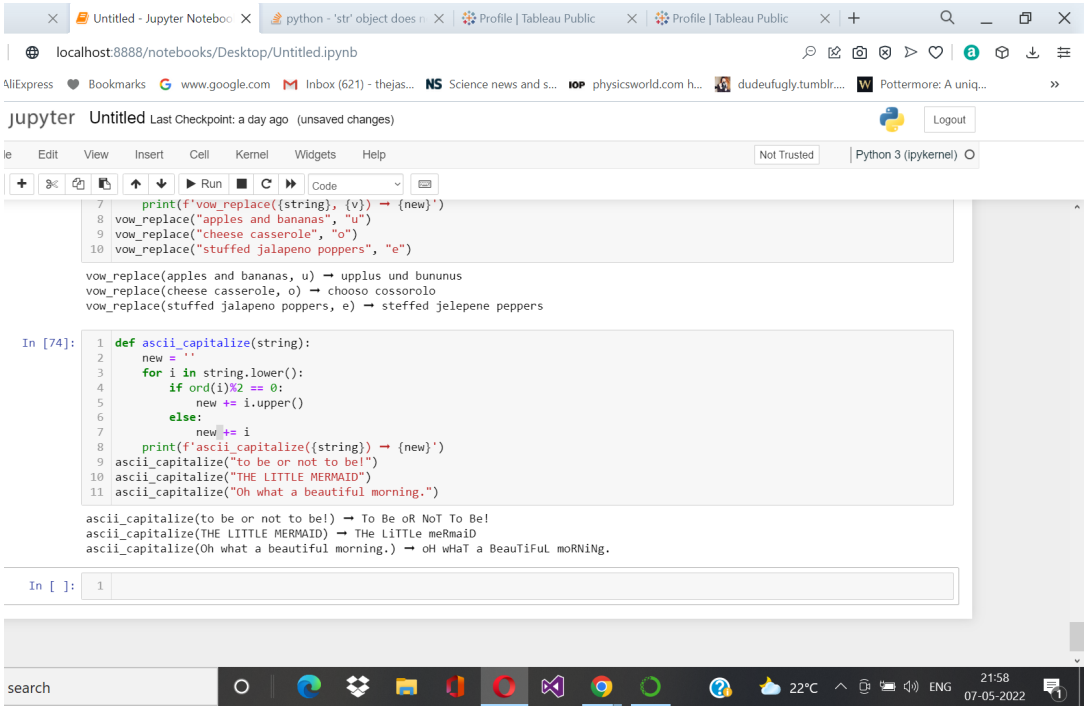
In [60]:
```
def vow_replace(string,v):
    vowels = ['a','e','i','o','u','A','E','I','O','U']
    new = string
    for i in string:
        if i in vowels:
            new = new.replace(i,v)
    print(f'vow_replace({string}, {v}) ➞ {new}')
vow_replace("apples and bananas", "u")
vow_replace("cheese casserole", "o")
vow_replace("stuffed jalapeno poppers", "e")
```

```
vow_replace(apples and bananas, u) ➞ upplus und bununus
vow_replace(cheese casserole, o) ➞ chooso cossorolo
vow_replace(stuffed jalapeno poppers, e) ➞ steffed jelepene peppers
```

5. Create a function that takes a string as input and capitalizes a letter if its ASCII code is even and returns its lower case version if its ASCII code is odd.

Examples:
ascii_capitalize("to be or not to be!") ➞ "To Be oR NoT To Be!"
ascii_capitalize("THE LITTLE MERMAID") ➞ "THe LiTTLe meRmaiD"
ascii_capitalize("Oh what a beautiful morning.") ➞ "oH wHaT a BeauTiFuL moRNiNg."

Ans:


Untitled - Jupyter Notebo
python - 'str' object does n
Profile | Tableau Public
Profile | Tableau Public
localhost:8888/notebooks/Desktop/Untitled.ipynb
AliExpress
Bookmarks
www.google.com
Inbox (621) - thejas...
Science news and s...
physicsworld.com h...
dudeufugly.tumblr....
Pottermore: A uniq...
jupyter
Untitled
Last Checkpoint: a day ago
(unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Trusted
Python 3 (ipykernel)
Run
Code
print(f'vow_replace({string}, {v}) → {new}')
vow_replace("apples and bananas", "u")
vow_replace("cheese casserole", "o")
vow_replace("stuffed jalapeno poppers", "e")
vow_replace(apples and bananas, u) → upplus und bununus
vow_replace(cheese casserole, o) → chooso cossorolo
vow_replace(stuffed jalapeno poppers, e) → steffed jelepene peppers
In [74]:
def ascii_capitalize(string):
new = ''
for i in string.lower():
if ord(i)%2 == 0:
new += i.upper()
else:
new += i
print(f'ascii_capitalize({string}) → {new}')
ascii_capitalize("to be or not to be!")
ascii_capitalize("THE LITTLE MERMAID")
ascii_capitalize("Oh what a beautiful morning.")
ascii_capitalize(to be or not to be!) → To Be oR NoT To Be!
ascii_capitalize(THE LITTLE MERMAID) → THe LiTTLe meRmaiD
ascii_capitalize(Oh what a beautiful morning.) → oH wHaT a BeauTiFuL moRNiNg.
In [ ]:
search
22°C
ENG
21:58
07-05-2022