

M2: Programming questions

1. `thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]`. 1) Check if "kiwi" exists in the list. 2) replace "melon" with "watermelon". 3) insert "blueberry" right after "cherry". 4) append "pear". 5) for `secondlist=['Peach', 'papaya']`, add it to `thislist` using the extend method.

2. `thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]`. Apply these functions or methods to `thislist`. 1) `del`; 2) `len`; 3) `remove`; 4) `clear`; 5) `pop`; 6) `sort`. Loop through `thislist`.

3. `thislist = ["apple", "banana", "cherry"]`

```
newlist = thislist
```

```
latestlist = thislist.copy()
```

```
newlist[[1]='orange'.
```

Show items in `thislist` and `latestlist` and explain why.

4. Run and see the differences.

```
thistuple = ("apple",)
print(type(thistuple))
```

#NOT a tuple. It is a string!

```
thistuple = ("apple")
print(type(thistuple))
```

5. `thistuple = ("apple", "banana", "cherry", "orange", "kiwi", "melon", "mango")`

Test: 1) `print(thistuple[-1])`

2) `print(thistuple[:4])`

3) `print(thistuple[-4:-1])`

6. Once a tuple is created, you cannot change its values. Tuples are **unchangeable**, or **immutable** as it also is called. But there is a workaround. You can convert the tuple into a list, change the list, and convert the list back into a tuple.

```
x = ("apple", "banana", "cherry")
y = list(x)
y[1] = "kiwi"
x = tuple(y)
```

```
print(x)
```

```
7. thisdict = {
    "brand": "Ford",
    "electric": False,
    "year": 1964,
    "colors": ["red", "white", "blue"]
}
```

show: 1) print (thisdict) 2) `print(len(thisdict))` 3) `print(thisdict['brand'])` 4) `print(str(thisdict.get("year")))` 5) `print(thisdict[0])`

8. thisset = {"apple", "banana", "cherry"}. Test: 1) thisset.remove('melon'); 2) thisset.discard('melon'); 3) thisset.pop(); 4) thisset.clear(); 4) del thisset. The del keyword will delete the set completely.

```
9. x = {"apple", "banana", "cherry"}
y = {"google", "microsoft", "apple"}
```

Show new items in z for: 1) `z = x.union(y)` 2) `z = x.intersection(y)` 3) `z=x.difference(y)` 4) `z=x.symmetric_difference(y)`

10. Run and understand the program [L8_count_file.py.](#) [Download L8_count_file.py.](#)

See [File.txt](#) [Download File.txt](#)

Group projects:

11. Write python code to create a password for a user name randomly and automatically.

The password should satisfy the conditions: (see sample code [pass_cr.py](#) [Download pass_cr.py](#))

- The length of the password should be from 8 to 15
- The first character should be a letter(upper case or lowercase)
- inclusion of one or more numerical digits ('0',..., '9')
- inclusion of one or more upper-case letters ('A',..., 'Z')
- inclusion of one or more lower-case letters ('a',..., 'z')
- inclusion of one or more special characters (@, #, \$, %, !)

12. `thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]`. 1) Based on a list of fruits, you want a new list, containing only the fruits with the letter "a" in the name. 2) Covert the first letters into upper cases for all items. Please write a program using list comprehensions to do them.