07. Programming Excercise (list, tuples, sets, dictionaries)

September 17, 2025

Excercise 01: Make a list of first ten letters of the alphabet, then using the slice operation do the following operations.

(a) Print the first three letters from the list

(b) Print any three letters from the middle i

(c) Print the letters from any particular index to the end of the list

[2]:
```
# create a list of first ten letters
alph_lst = ["a", "b", "c", "d", "e", "f", "g", "h", "i", "j" ]
```

[6]:
```
# Print the first three letters from the list

print(alph_lst[0:3]) #Way 01
```

```
['a', 'b', 'c']
```

[11]:
```
for i in range(0,3):
    print(alph_lst[i], end=", ")
```

```
a, b, c,
```

[12]:
```
#  Print any three letters from the middle i
alph_lst[4:7]
```

[12]:
```
['e', 'f', 'g']
```

[13]:
```
#  Print the letters from any particular index to the end of the list
alph_lst[4: ]
```

[13]:
```
['e', 'f', 'g', 'h', 'i', 'j']
```

Excercise 02: Write a program using reduce () function to calculate the sum of first 10 natural numbers

[1]:
```
from functools import reduce
def add(num1, num2):
    return num1 + num2
```

```python
num_lst = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

print("Sum of the values in list: ")
print(reduce(add, num_lst))
```

```
Sum of the values in list:
55
```

Excercise 03: Write a program that creates a list [“a”, “b”, “c”], then create a tuple from that list. Now do the opposite. That is, create the tuple(“a”, “b”, “c”) and then create a list from it

[2]:

```python
lst = ["a", "b", "c"]

#create a tuple from that list
tple = tuple(lst)

#create a list from that tuple
lst = list(tple)
```

[4]:

```python
tple
```

[4]:

```
('a', 'b', 'c')
```

[5]:

```python

```

[5]:

```
['a', 'b', 'c']
```

[6]:

```python
lst
```

[6]:

```
['a', 'b', 'c']
```

Excerise 04: Create a tuple that has just one element which in turn may have three elements “a”, “b”, “c”. Print the length of this tuple

[7]:

```python
tple = tuple("abc")
```

[8]:

```python
tple
```

[8]:

```
('a', 'b', 'c')
```

[9]:

```python
len(tple)
```

[9]:

```
3
```

Excercise 05: Create a dictionary of products purchased and their MRPs. Calculate the bill and display to the customer

[21]:

```python
product = {"laptop": 50000, "mouse": 2000, "Keyboard": 5000}
sum = 0
for key, value in product.items():
    sum += value

print("Product bill : ", sum)
```

```
Product bill :  57000
```

Excercise 05: Create an empty dictionary named captains.

[1]:
```
#create empty dictionary
captains = {}
```

[2]:
```
captains
```

[2]:
```
{}
```

Excercise 06: Using the square bracket notation, enter the following data into the dictionary, one item at a time: ‘Enterprise’: ‘Picard’ ‘Voyager’: ‘Janeway’ ‘Defiant’: ‘Sisko’

[11]:
```
dic = {
'Enterprise': 'Picard',
'Voyager': 'Janeway',
'Defiant': 'Sisko'
}
```

[5]:
```
dic
```

[5]:
```
{'Enterprise': 'Picard', 'Voyager': 'Janeway', 'Defiant': 'Sisko'}
```

[13]:
```
if "Enterprise" in dic:
    print(f"The value of Enterprise is {dic['Enterprise']}")
else:
    dic["Enterprise"] = "Unknown"

if "Discovery" in dic:
     print(f"The value of Enterprise is {dic['Enterprise']}")
else:
    dic["Discovery"] = "Unknown"
```

```
The value of Enterprise is Picard
```

Excercise 07: Write a for loop to display the ship and captain names contained in the dictionary. For example, the output should look something like this: The Enterprise is captained by Picard.

[14]:
```
for ship_name, captain_name in dic.items():
    print(f"The {ship_name} is captained by {captain_name} ")
```

```
The Enterprise is captained by Picard
The Voyager is captained by Janeway
The Defiant is captained by Sisko
The Discovery is captained by Unknown
```

```
[15]: # delete "Discovery" from the dictionary
      del dic["Discovery"]
```

```
[16]: dic
```

```
[16]: {'Enterprise': 'Picard', 'Voyager': 'Janeway', 'Defiant': 'Sisko'}
```

```
[18]: key_value_pairs = (
          ('Enterprise', 'Picard'),
          ('Voyager', 'Janeway'),
          ('Defiant', 'Sisko'),
      )
```

```
[19]: ships = dict(key_value_pairs)
```

```
[20]: ships
```

```
[20]: {'Enterprise': 'Picard', 'Voyager': 'Janeway', 'Defiant': 'Sisko'}
```

```
[ ]:
```