

Name: Afrida Islam

Student ID: 21221030

✓ EEE385IL (MACHINE LEARNING LABORATORY)

LAB 2: Introduction to Python

List in Python

Lists are used to store multiple items in a single variable.

```
list_name = [val_1, val_2, val_3, .....]
```

✓ Common List operations

```
marks = [88, 56, 94, 82, 94]
print("Before changing any element      --> ", marks)
```

```
marks.append(99)
print("After appending one element      --> ", marks)
```

```
marks.remove(94)
print("After removing one element      --> ", marks)
```

```
marks.insert(1, 71)
print("After inserting '71' at index '1' --> ", marks)
```

```
marks.pop(2)
print("After popping out the element at index '2' --> ", marks)
```

```
➦ Before changing any element      --> [88, 56, 94, 82, 94]
  After appending one element      --> [88, 56, 94, 82, 94, 99]
  After removing one element      --> [88, 56, 82, 94, 99]
  After inserting '71' at index '1' --> [88, 71, 56, 82, 94, 99]
  After popping out the element at index '2' --> [88, 71, 82, 94, 99]
```

```
my_list = [10, 'a', "Hello", 20.56, 5.78, 'b']
print(my_list)
print(my_list[2:5:1])
```

```
➦ [10, 'a', 'Hello', 20.56, 5.78, 'b']
  ['Hello', 20.56, 5.78]
```

```
my_list = [[10,20,30], 40, [50,60]]
print(my_list[2][1])
```

```
➦ 60
```

```
my_list = [[[10,20.5,'a']]]
print(my_list[0][0][2])
```

```
➦ a
```

✓ TASK (Practice)

Create a list containing the names of the cars:

- Toyota Corolla
- Mazda RX-8
- Lamborghini Gallardo

Then insert a new car 'Mercedes Benz' at the 2nd position of the list.

```
cars = ['Toyota Corolla', 'Mazda RX-8', 'Lamborghini Gallardo']
print(cars)
```

```

for car in cars:
    print(car)

cars.insert(1, 'Mercedes Benz')
print(cars)

type(cars)

➦ ['Toyota Corolla', 'Mazda RX-8', 'Lamborghini Gallardo']
Toyota Corolla
Mazda RX-8
Lamborghini Gallardo
['Toyota Corolla', 'Mercedes Benz', 'Mazda RX-8', 'Lamborghini Gallardo']
list

```

▼ Tuples

Tuples in Python contain multiple values that are ordered, indexed but **unchangable**.

```

cars = ('Toyota Corolla', 'Mazda RX-8', 'Lamborghini Gallardo')
print(cars[0])

marks = (86, 96, 78, 56)
print(marks[0])

mixed_tup = ('Toyota Corolla', 160)
print(mixed_tup)

type(cars)

➦ Toyota Corolla
86
('Toyota Corolla', 160)
tuple

print(type(cars))
cars_l = list(cars)
print(type(cars_l))
cars_l.append('Tesla')
print(cars_l)

➦ <class 'tuple'>
<class 'list'>
['Toyota Corolla', 'Mazda RX-8', 'Lamborghini Gallardo', 'Tesla']

```

▼ Set

Sets in python can contain multiple values and are **unordered, unindexed** and **unchangable**.

```

cars = {'Toyota Corolla', 'Mazda RX-8', 0}
print(cars)
type(cars)

➦ {0, 'Toyota Corolla', 'Mazda RX-8'}
set

```

▼ Dictionary in Python

Python's dictionaries allow you to connect pieces of related information. Each piece of information in a dictionary is stored as a **key-value pair**. When you provide a **key**, Python returns the **value** associated with that key. You can loop through all the key-value pairs, all the keys, or all the values.

```

cars = {'brand': ['Toyota', 'Mazda', 'Lamborghini'],
        'model': ['Corolla', 'RX-8', 'Gallardo'],
        'year': [1976, 1984, 1998]}

print(cars.items())
print(cars.keys())
print(cars.values())

# for key in cars.keys():
#     print(key)

# m = cars['model']
# print(m)

```

```
#for y in ys:
    #print(y)

cars['speed'] = [150, 180, 220]
print(cars.items())
cars['Mileage'] = [10,6,3]
print(cars.items())

# del cars['year']
# print(cars.items())

dict_items([('brand', ['Toyota', 'Mazda', 'Lamborghini']), ('model', ['Corolla', 'RX-8', 'Gallardo']), ('year', [1976, 1984, 1998])])
dict_keys(['brand', 'model', 'year'])
dict_values(['Toyota', 'Mazda', 'Lamborghini'], ['Corolla', 'RX-8', 'Gallardo'], [1976, 1984, 1998])
dict_items([('brand', ['Toyota', 'Mazda', 'Lamborghini']), ('model', ['Corolla', 'RX-8', 'Gallardo']), ('year', [1976, 1984, 1998])]),
dict_items([('brand', ['Toyota', 'Mazda', 'Lamborghini']), ('model', ['Corolla', 'RX-8', 'Gallardo']), ('year', [1976, 1984, 1998])])
```

✓ TASK-1

Create a dictionary with the "key"s as "ID", "Name", "credits" and input the corresponding information of three students.
Append another "key" named "cgpa" after that.

```
#####
##### code starts here #####
keys = {'ID': [21221030,21221031,21221032],
        'Name': ['A', 'B', 'C'],
        'Credits': [111,30, 78]}
keys['cgpa'] = [3,2.6,3.8]
print(keys.items())

##### code ends here #####
#####

dict_items([('ID', [21221030, 21221031, 21221032]), ('Name', ['A', 'B', 'C']), ('Credits', [111, 30, 78]), ('cgpa', [3, 2.6, 3.8])])

cars = {'brand': ['Toyota', 'Mazda', 'Lamborghini'],
        'model': ['Corolla', 'RX-8', 'Gallardo'],
        'year': [1976, 1984, 1998]}

for k, v in cars.items():
    print(k)
    for val in v:
        print(val)
```

```
brand
Toyota
Mazda
Lamborghini
model
Corolla
RX-8
Gallardo
year
1976
1984
1998
```

✓ Function

```
def function_name(input_args):

    statements

    return output_var
```

```
def easy_calc_add(n1, n2):
    return n1 + n2

a = 'Helloooo'
b = ' Afrida'
result = easy_calc_add(a, b)
print(result)
```

➞ Helloooo Afrida

```
def welcome_msg(names):
    for name in names:
        print("Welcome {}".format(name))
```

```
people = ['ABC', 'XYZ', 'PQR']
welcome_msg(people)
```

➞ Welcome ABC!!
Welcome XYZ!!
Welcome PQR!!

```
import math
def easy_calc_log(n1, eps = 0.000001):
    if n1 < 0:
        print("MATH ERROR! Log is not defined for negative numbers!")
        return

    return math.log10(n1 + eps)
```

```
number = 10
print(easy_calc_log(number, 0))
```

➞ 1.0

```
import math
def calc_log10(n):
    if n > 0:
        return math.log10(n)
    else:
        print("UNDEFINED!!")
```

```
print(calc_log10(0))
```

➞ UNDEFINED!!
None

```
def std_info(ID, name = 'Bracuian'):
    print("Welcome! {}: {}".format(ID, name))
```

```
id = 28321087
std_info(id, 'ABC')
```

➞ Welcome! 28321087: ABC!

✓ TASK-2

Write a function in Python to determine whether a number is Armstrong or not.

```
#####
##### code starts here #####
import math

def isarmstrong(n):
    m = n
    sum = 0

    while 1:
        dig = n % 10
        n = int(n / 10)
        sum = sum + math.pow(dig, 3)

        if n == 0:
            break

    if sum == m:
        print("ARMSTRONG number!!")
    else:
        print("NOT a Armstrong number!")

##### code ends here #####
```

➡ ARMSTRONG number !!

Brand: Toyota
Model: Corolla
Year: 1978

Brand: Toyota
Model: Premio
Year: 1978

```
class Car():
    """ Description of the Car class """

    def __init__(self, brand, model, year):
        """Initializaing the attributes if a car object"""
        self.brand = brand
        self.model = model
        self.year = year

    def show_info(self):
        print('Brand: {} \nModel: {} \nYear: {} \n'.format(self.brand, self.model, self.year))

    def update_model(self, new_model):
        self.model = new_model
```

```
first_car = Car('Toyota', 'Corolla', 1978)
first_car.show_info()
print(first_car.brand)
```

```
second_car = Car('Mazda', 'RX-8', 1996)
second_car.show_info()
second_car.update_model('RX-7')
second_car.show_info()
```

Brand: Toyota
Model: Corolla
Year: 1978

Toyota
Brand: Mazda
Model: RX-8
Year: 1996

Brand: Mazda
Model: RX-7
Year: 1996

```
# Install the package for Tex and then convert to PDF directly as LaTeX
!sudo apt-get install texlive-xetex texlive-fonts-recommended texlive-plain-generic
```

```
# Provide the file path of the notebook file
```

```
!jupyter nbconvert --to pdf "/content/drive/MyDrive/Colab Notebooks/Afrida Islam_Lab02_21221030.ipynb"
```



Show hidden output

```
from google.colab import drive  
drive.mount('/content/drive')
```



Show hidden output