ASSIGNMENT - 1

Python Methods

1) enumerate ( )

→ The 'enumerate ( )' function is used to iterate over a sequence such as a list, tuple or string along with an index, which represents the position of each element in the sequence

Syntax :- enumerate (iterable, start=0)

- iterable : The sequence to be iterated
- start : (optional) integer that specifies starting index for enumeration.

Example :-

```
fruits = ['apple', 'banana', 'cherry', 'date']

for index, fruits in enumerate (fruits):
    print (f"Index {index}: {fruits}")
```

assignment > enum.py > ...

```python
fruits = ['apple', 'banana', 'cherry', 'date']
for index, fruit in enumerate(fruits):
    print(f"Index {index}: {fruit}")

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\vaishnavi\python> & C:/Users/Shree/AppData/Local/Programs/Python/Python311/python.exe d:/vaishnavi/python/assignment/enum.py
Index 0: apple
Index 1: banana
Index 2: cherry
Index 3: date
```

Python

Python

2) reduce( )

→ The 'reduce( )' function is a part of the 'functools' module.

→ It is used to apply a specified binary function cumulatively to the items of an iterable, from left to right.

→ It is used to reduce the iterable to a single accumulated result.

Syntax :- functools.reduce(function, iterable, initial)

Example :-

```
from functools import reduce
numbers = [1,2,3,4, 5]
def multiply (x,y):
    return x*y
```

```
product = reduce (multiply, numbers)
print (product)
```

o/p => 120

assignment > reduces.py > multiply

```python
from functools import reduce
numbers = [1, 2, 3, 4, 5]
def multiply(x, y):
    return x * y

product = reduce(multiply, numbers)
print(product)

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\vaishnavi\python> & C:/Users/Shree/AppData/Local/Programs/Python/Python311/python.exe d:/vaishnavi/python/assignment/reduces.py
120
```

3) map ()

→ In python, 'map()' function is a built-in function that is used to apply a specified function to all items in an iterable and return a new iterable containing the results.

Syntax: map (function, iterable)

- function: It is the function that you want to apply to each element in the iterable.
- iterable: This is the iterable whose elements you want to transform

Example:-

```
def square (x):
    return x**2

num = [1,2,3,4,5]

sq_num = map(square, num)
res = list (sq_num)
print (res)
```

O/P ⇒ [1, 4, 9, 16, 25]

assignment > map.py > ...

```python
def square(x):
    return x ** 2
num = [1, 2, 3, 4, 5]
sq_num = map(square, num)
res = list(sq_num)
print(res)
```

PROBLEMS OUTPUT TERMINAL ··· Python

```
PS D:\vaishnavi\python> & C:/Users/Shree/AppData/Local/Programs
/Python/Python311/python.exe d:/vaishnavi/python/assignment/map
.py
[1, 4, 9, 16, 25]
```

4) <u>filter()</u>

→ It is a built-in function that is used to filter elements from an iterable based on a specific function or condition.

<u>Syntax</u>:- filter(function, iterable)

Example :-

```
def is_even(x):
    return x % 2 == 0
numbers = [1,2,3,4,5,6,7,8,9,10]
even = filter (is_even, numbers)
res = list (even)
print (res)
```

o/p :→ [2, 4, 6, 8, 10]

assignment > filter.py > ...

```python
def is_even(x):
    return x % 2 == 0
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
even = filter(is_even, numbers)
res = list(even)
print(res)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python

```
PS D:\vaishnavi\python> & C:/Users/Shree/AppData/Local/Programs/Python/Python311/
python.exe d:/vaishnavi/python/assignment/filter.py
[2, 4, 6, 8, 10]
```

5) zip ()

→ 'zip()' function is a built-in function
→ It is used to combine multiple iterables into a single iterable
→ It creates a new iterable where each element is a tuple containing elements from the input iterables at the same position.

Syntax:- zip (iterable1, iterable2, ---)

Example:-

```
names = ["Alice", "Bob", "Charlie"]
scores = [95, 82, 78]
paired_data = zip (names, scores)
result_list = list (paired_data)
print (result_list)
```

O/P:- [('Alice', 95), ('Bob', 82), ('charlie', 78)]

assignment > zip.py > ...

```python
names = ["Alice", "Bob", "Charlie"]
scores = [95, 82, 78]
paired_data = zip(names, scores)
result_list = list(paired_data)
print(result_list)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python

```
PS D:\vaishnavi\python> & C:/Users/Shree/AppData/Local/Programs/Python/Python
311/python.exe d:/vaishnavi/python/assignment/zip.py
[('Alice', 95), ('Bob', 82), ('Charlie', 78)]
```

6) id( )

→ 'id( )' function is a built-in function
→ It returns the unique identity of an object
→ This identity is a unique & constant value associated with the object during its life time.

Syntax :- id (object)

Example :-

```
x = 42
y = x
id_x = id(x)
id_y = id(y)
print (id_x)
print (id_y)
if id_x == id_y :
    print ("x & y reference the same object")
else :
    print ("x and y reference different objects")
```

o/p :- ~~seconday~~ .

```
140717962684488
140717962684488
x and y reference the same object.
```

assignment > id.py > ...

```python
x = 42
y = x
id_x = id(x)
id_y = id(y)
print(id_x)
print(id_y)
if id_x == id_y:
    print("x and y reference the same object")
else:
    print("x and y reference different objects")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python

```
PS D:\vaishnavi\python> & C:/Users/Shree/AppData/Local/Programs/Python/Python
311/python.exe d:/vaishnavi/python/assignment/id.py
140717962684488
140717962684488
x and y reference the same object
```