

CONFIDENTIAL & RESTRICTED

Gaurav Gupta

Sequence Type List

- List Creation
- List Mutability
- Operations
- Slicing

tuteur.py@gmail.com

CONFIDENTIAL & RESTRICTED

Gaurav Gupta

List

- **[1,2,3, True, 'abcd']**
- **Mutable Sequence** type with elements separated by a comma.

```
l1 = []
```

```
l2 = list()
```

```
l3 = [1,2,3]
```

```
l4 = list(l3)
```

```
l5 = list('string')
```

tuteur.py@gmail.com

CONFIDENTIAL & RESTRICTED

Gaurav Gupta

List

- **Mutability**

- `l [1] = 4`

- `l.append(5)`

- `l.insert(2,33)`

- `l.extend([10 ,20])`

- `len(l)`

- **WAP** to input a sentence from user , and print one random word out of it.

tuteur.py@gmail.com

CONFIDENTIAL & RESTRICTED

Gaurav Gupta

List Functions

- **In Place** operations

- `l.sort()`

- `l.index()`

- `l.pop()`

- `l.remove()`

- Indexing:

- `l = [[10, 20], [True, False], [], 'abcd']`

- `l [0] [1]`

- `l [3] [3]`

tuteur.py@gmail.com

CONFIDENTIAL & RESTRICTED

Gaurav Gupta

Sequence Type Tupe

- Tuple Creation
- Immutability
- Operations
- Slicing

tuteur.py@gmail.com

CONFIDENTIAL & RESTRICTED

Gaurav Gupta

Tuple

(1 ,2.3 , True, 'ABCD')

- **Immutable** sequences. Represented by a **()**
- - `x = ()`
 - `x = tuple()`
 - `x = (1,2,3)`
 - `x = 1,2,3`
 - `x = 1,`
 - `x = tuple([1,2,3])`

tuteur.py@gmail.com

CONFIDENTIAL & RESTRICTED

Gaurav Gupta

Tuple

Modifications not allowed

```
x = (1, 2, 3)
x[1] = 3
```

tuteur.py@gmail.com

CONFIDENTIAL & RESTRICTED

Gaurav Gupta

Copying Lists

- Simple assignments don't create copy
l2 = l1 # both are same
- Copying requires special call to **list()** or **slicing**
l2 = list(l1)
l2 = l1 [:]
l2 = l2 [::]

tuteur.py@gmail.com

CONFIDENTIAL & RESTRICTED

Gaurav Gupta

Common operations on Sequences

- **len()** : returns the number of elements
- Slicing.
- Membership check

in , not in # returns Boolean **True** or **False**

- Finding minimum and maximum values:

min, max

- Concatenation and Replication

+, *

tuteur.py@gmail.com

CONFIDENTIAL & RESTRICTED

Gaurav Gupta

Loops

- While Loop
- Break and continue
- List Comprehension

tuteur.py@gmail.com

CONFIDENTIAL & RESTRICTED

Gaurav Gupta

While Loop

- Syntax :
while *<condition>*:
 statements1
 else: *# optional*
 statements2
- *Statements2* is executed when condition becomes false (but not in case of break)
- WAP to print first 10 natural numbers. Update the program to print their sum
- WAP to count vowels in a string input by user.
- WAP to print all multiples of **3** till **N** (input N from user).

tuteur.py@gmail.com

CONFIDENTIAL & RESTRICTED

Gaurav Gupta

Break and Continue

- **break** statement is used to terminate the current loop
- On execution, **continue** statement skips the statements below it in the current loop and forces next iteration of the loop.
- Update the **rolling dice** program to ask user to roll again or exit(break).
- Update the **rolling dice** program to also check for invalid inputs(continue)

tuteur.py@gmail.com

CONFIDENTIAL & RESTRICTED

Gaurav Gupta

List Comprehension : For loop

- Syntax:
[expression(<variable>) **for** <variable> **in** <sequence type> [if <condition>]]
condition is optional
- WAP to generate list of first 10 natural numbers (Generate a list of their squares also).
- WAP to count vowels using list comprehension
- WAP to find sum of the squares of first 10 even numbers
4 + 9 + 16 + 25

tuteur.py@gmail.com