TRIBHUVAN UNIVERSITY

Institute of Engineering

Pulchowk Campus

A LAB REPORT ON

Assignment 1

LAB NO. :

EXPERIMENT DATE : 2025-06-28

SUBMITTED DATE : 2025-07-01

SUBMITTED BY

Name : Abhishek Kharel

Group : A

Roll No. : 081BEL005

SUBMITTED TO

Department of

Computer Engineering

Python Built-in Methods Assignment

Page 1: Student Details

Name: Abhishek Kharel
Roll No: 081BEL005
Date: July 1, 2025

String Methods

name of method: upper

- **Syntax:** string.upper()
- **Usage (code):**

```
text = "hello world"
upper_text = text.upper()
print(upper_text)
```

- **output:** HELLO WORLD

name of method: lower

- **Syntax:** string.lower()
- **Usage (code):**

```
text = "HELLO WORLD"
lower_text = text.lower()
print(lower_text)
```

- **output:** hello world

name of method: strip

- **Syntax:** string.strip([chars])
- **usage (code):**

```
text = "    some spaces    "
stripped_text = text.strip()
print(f"'{stripped_text}'")
```

- **output:** 'some spaces'

name of method: split

- **Syntax:** string.split(separator, maxsplit)
- **Usage (code):**

```
text = "apple,banana,cherry"
fruits = text.split(',')
print(fruits)
```

- **output:** ['apple', 'banana', 'cherry']

name of method: join

- **Syntax:** string.join(iterable)
- **Usage (code):**

```
words = ['Hello', 'World']
sentence = ' '.join(words)
print(sentence)
```

- **output:** Hello World

name of method: replace

- **Syntax:** string.replace(oldvalue, newvalue, count)
- **Usage (code):**

```
text = "I like cats."
new_text = text.replace("cats", "dogs")
print(new_text)
```

- **output:** I like dogs.

name of method: find

- **Syntax:** string.find(value, start, end)
- **Usage (code):**

```
text = "Hello, welcome to my world."
position = text.find("welcome")
print(position)
```

- **output:** 7

List Methods

name of method: append

- **Syntax:** list.append(element)
- **Usage (code):**

```
fruits = ['apple', 'banana']
fruits.append('cherry')
print(fruits)
```

- **output:** ['apple', 'banana', 'cherry']

name of method: extend

- **Syntax:** list.extend(iterable)
- **Usage (code):**

```
list1 = [1, 2, 3]
list2 = [4, 5, 6]
list1.extend(list2)
print(list1)
```

- **output:** [1, 2, 3, 4, 5, 6]

name of method: insert

- **Syntax:** list.insert(position, element)
- **Usage (code):**

```
fruits = ['apple', 'cherry']
fruits.insert(1, 'banana')
print(fruits)
```

- **output:** ['apple', 'banana', 'cherry']

name of method: pop

- **Syntax:** list.pop(position)
- **Usage (code):**

```
numbers = [1, 2, 3, 4]
popped_element = numbers.pop(2)
```

```
print(f"Popped: {popped_element}, Remaining List: {numbers}")
```
- **output:** Popped: 3, Remaining List: [1, 2, 4]

name of method: sort

- **Syntax:** list.sort(reverse=True|False, key=myFunc)
- **Usage (code):**

```
numbers = [3, 1, 4, 2]
numbers.sort()
print(numbers)
```
- **output:** [1, 2, 3, 4]

Tuple Methods

Tuples are immutable, so they have very few methods.

name of method: count

- **Syntax:** tuple.count(value)
- **Usage (code):**

```
my_tuple = (1, 2, 3, 2, 4, 2)
count_of_2 = my_tuple.count(2)
print(count_of_2)
```
- **output:** 3

name of method: index

- **Syntax:** tuple.index(value)
- **Usage (code):**

```
my_tuple = ('a', 'b', 'c', 'd')
index_of_c = my_tuple.index('c')
print(index_of_c)
```
- **output:** 2

Set Methods

name of method: add

- **Syntax:** set.add(element)
- **Usage (code):**

```
my_set = {1, 2, 3}
my_set.add(4)
print(my_set)
```
- **output:** {1, 2, 3, 4}

name of method: update

- **Syntax:** set.update(iterable)
- **Usage (code):**

```
set1 = {'a', 'b'}
set2 = ['b', 'c']
set1.update(set2)
print(set1)
```
- **output:** {'a', 'b', 'c'}

name of method: remove

- **Syntax:** set.remove(element)
- **Usage (code):**

```
my_set = {1, 2, 3}
my_set.remove(2)
print(my_set)
```

- **output:** {1, 3}

name of method: union

- **Syntax:** set.union(set1, set2...)
- **Usage (code):**

```
set1 = {1, 2, 3}
set2 = {3, 4, 5}
union_set = set1.union(set2)
print(union_set)
```

- **output:** {1, 2, 3, 4, 5}

name of method: intersection

- **Syntax:** set.intersection(set1, set2...)
- **Usage (code):**

```
set1 = {1, 2, 3}
set2 = {3, 4, 5}
intersection_set = set1.intersection(set2)
print(intersection_set)
```

- **output:** {3}

Dictionary Methods

name of method: keys

- **Syntax:** dictionary.keys()
- **Usage (code):**

```
person = {'name': 'Abhishek', 'age': 20}
keys = person.keys()
print(keys)
```

- **output:** dict_keys(['name', 'age'])

name of method: values

- **Syntax:** dictionary.values()
- **Usage (code):**

```
person = {'name': 'Abhishek', 'age': 20}
values = person.values()
print(values)
```

- **output:** dict_values(['Abhishek', 20])

name of method: items

- **Syntax:** dictionary.items()
- **Usage (code):**

```
person = {'name': 'Abhishek', 'age': 20}
```

```
items = person.items()
print(items)
```

- **output:** dict_items([('name', 'Abhishek'), ('age', 25)])

name of method: get

- **Syntax:** dictionary.get(keyname, value)
- **usage (code):**

```
person = {'name': 'Alice', 'age': 25}
name = person.get('name')
print(name)
```

- **output:** Alice

name of method: update

- **Syntax:** dictionary.update(iterable)
- **usage (code):**

```
person = {'name': 'Abhishek', 'age': 20}
person.update({'city': 'New Baneshwor'})
print(person)
```

- **output:** {'name': 'Abhishek', 'age': 20, 'city': 'New Baneshwor'}