

# exp\_2

February 17, 2025

## 1 Experiment 2: Basic elements of Python

- Name: Anas Muhammed Sahil
- Date: 16-01-2025
- Roll Number: 20242AIE0010

```
[1]: # Implement basics elements of Python namely Tuples, Funcs
# Dicts, exceptions, assertions.

# Tuples
# Tuples are immutable, ordered collection of elements

def tuple_example():
    # Create a tuple
    t = (1, 2, 3)
    print(t)

    # Access elements of a tuple
    print(t[0])
    print(t[1])
    print(t[2])

    # Tuples are immutable
    # t[0] = 100 # This will raise an error

    # Tuple with one element
    t = (1,) # Note the comma
    print(t)

    # Tuple unpacking
    x, y, z = (1, 2, 3)
    print(x)
    print(y)
    print(z)

    # Tuple unpacking with *
    x, *y, z = (1, 2, 3, 4, 5)
    print(x)
```

```

print(y)
print(z)

# Tuple methods
t = (1, 2, 3, 4, 5)
print(t.count(1))
print(t.index(1))

def list_example():
    # Create a list
    l = [1, 2, 3]
    print(l)

    # Access elements of a list
    print(l[0])
    print(l[1])
    print(l[2])

    # Lists are mutable
    l[0] = 100
    print(l)

    # List with one element
    l = [1]
    print(l)

    # List methods
    l = [1, 2, 3, 4, 5]
    l.append(6)
    print(l)
    print(l.count(1))
    l.extend([7, 8, 9])
    print(l)
    print(l.index(1))
    l.insert(0, 0)
    print(l)
    print(l.pop())
    print(l)
    l.remove(0)
    print(l)
    l.reverse()
    print(l)
    l.sort()
    print(l)

def dict_example():

```

```

# Create a dictionary
d = {'name': 'John', 'age': 25}
print(d)

# Access elements of a dictionary
print(d['name'])
print(d['age'])

# Dictionary methods
print(d.keys())
print(d.values())
print(d.items())
print(d.get('name'))
print(d.pop('name'))
print(d)
d.update({'name': 'John'})
print(d)

# Dictionary unpacking
d = {'name': 'John', 'age': 25}
print(d)
# print(**d)

def advanced_lists():
    # List comprehension
    l = [i for i in range(10)]
    print(l)

    # List comprehension with condition
    l = [i for i in range(10) if i % 2 == 0]
    print(l)

    # Nested list comprehension
    l = [[i for i in range(3)] for j in range(3)]
    print(l)

def exception_example():
    try:
        x = 1 / 0
    except ZeroDivisionError as e:
        print(e)
    except Exception as e:
        print(e)
    finally:
        print('Finally block')

```

```
def assert_example():
    x = 1
    assert x == 1
    assert x == 2, 'x should be 2'

if __name__ == '__main__':
    tuple_example()
    list_example()
    dict_example()
    # assert_example()
    advanced_lists()
```

```
(1, 2, 3)
1
2
3
(1,)
1
2
3
1
[2, 3, 4]
5
1
0
[1, 2, 3]
1
2
3
[100, 2, 3]
[1]
[1, 2, 3, 4, 5, 6]
1
[1, 2, 3, 4, 5, 6, 7, 8, 9]
0
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
9
[0, 1, 2, 3, 4, 5, 6, 7, 8]
[1, 2, 3, 4, 5, 6, 7, 8]
[8, 7, 6, 5, 4, 3, 2, 1]
[1, 2, 3, 4, 5, 6, 7, 8]
{'name': 'John', 'age': 25}
John
25
dict_keys(['name', 'age'])
dict_values(['John', 25])
dict_items([('name', 'John'), ('age', 25)])
```

```
John
John
{'age': 25}
{'age': 25, 'name': 'John'}
{'name': 'John', 'age': 25}
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
[0, 2, 4, 6, 8]
[[0, 1, 2], [0, 1, 2], [0, 1, 2]]
```

[ ]: