

NAME: Muhammad Johar Shah

ID: 14124

BSSE

Subject: Data Sciences

Submitted To: Sir Ayub

Ans no.1:

There are many types of errors in Python. Few of them are listed below.

TypeError is thrown when an operation or function is applied to an object of an inappropriate type.

```
>>> '2'+2
```

Traceback (most recent call last):

File "<pyshell#23>", line 1, in <module>

```
'2'+2
```

TypeError: must be str, not int

ValueError is thrown when a function's argument is of an inappropriate type.

```
>>> int('xyz')
```

Traceback (most recent call last):

File "<pyshell#14>", line 1, in <module>

```
int('xyz')
```

ValueError: invalid literal for int() with base 10: 'xyz'

NameError is thrown when an object could not be found.

```
>>> age
```

Traceback (most recent call last):

File "<pyshell#6>", line 1, in <module>

age

NameError: name 'age' is not defined

ZeroDivisionError is thrown when the second operator in the division is zero.

```
>>> x=100/0
```

Traceback (most recent call last):

File "<pyshell#8>", line 1, in <module>

```
x=100/0
```

ZeroDivisionError: division by zero

IndexError is thrown when trying to access an item at an invalid index.

```
>>> L1=[1,2,3]
```

```
>>> L1[3]
```

Traceback (most recent call last):

File "<pyshell#18>", line 1, in <module>

```
L1[3]
```

IndexError: list index out of range

Ans no.2:

A string in Python can be Boolean tested for truth value.

The return type will be in Boolean value (True or False)

Example:

```
my_string="Hello World"
```

```
print my_string.isalnum()
```

```
#False
```

```
print my_string.isalpha()
```

```
#False
```

```
print my_string.isdigit()
```

```
#False
```

```
print my_string.istitle()
```

```
#True
```

```
print my_string.isupper()
```

```
#False
```

```
print my_string.islower()
```

```
#False
```

```
print my_string.isspace()
```

```
#False
```

```
print my_string.endswith('d')
```

```
#True
```

```
print my_string.startswith('H')
```

```
#True
```

Ans no:3

Formatted string inputs are used to place variables inside a string. Usually strings are concatenated using the “+” sign. But in formatted strings, we use placeholders for different variables for different occasions.

Example:1

```
#CODE in Python 3.6
#not using formatted string

name = "johar"

Age = 25

print("My name is " + name + "and my age is " + str(age))
```

Example:2

```
#CODE in Python 3.6
#using formatted strings

name = "johar"

Age = 25

print(f"My name is {name} and my age is {age}")
```