

Exceptions

Exception is an event which occurs during the execution of the program that disrupt the normal flow of programs.

Example:

| Exception name | Description |
|-------------------|--|
| Exception | Base class for all exceptions. |
| TypeError | Raised when you try to operation that does not belong to that datatype. |
| NameError | Raised when you enter the variable name wrong or you have not declared it. |
| ZeroDivisionError | Raised when you try to divide or modulus any number by 0. |
| IndexError | Raised when index is not in sequence. |
| SyntaxError | Raised when there is some syntax error in code. |
| IndentationError | Raised when indentation is not written properly. |
| KeyError | Raised when you try to access any key that is not in dictionary. |

Try ... except

Syntax:

```
try:  
    operations  
except:  
    if you get any error execute this code.
```

Example 1:

```
try:  
    var = [1,2,3,4,5]  
    print(var[10])  
except:  
    print('some error took place')
```

Output:

some error took place

Example 2:

```
try:
    var = [1,2,3,4,5]
    print(var[10])
except IndexError:
    print('index error in the code')
except:
    print('some error in code')
```

Output:

index error in the code

Example 2:

```
try:
    var = [1,2,3,4,5]
    print(var[10])
except:
    print('some error in code')
except IndexError:
    print('index error in the code')
```

Output:

some error in code

Try...except as arguments:

Syntax:

```
try:  
    operations  
except as arg_var:  
    if you get any error execute this code.
```

Example:

```
try:  
    var = [1,2,3,4,5]  
    print(var[10])  
except IndexError as arg:  
    print('some error in code')  
    print(arg)
```

Output:

```
some error in code  
list index out of range
```

Try except finally

Syntax:

try:

operations

except:

if you get any error execute this code.

finally:

#execute this code finally either exception is raised or not

Example:

try:

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

 print(var[10])

except:

 print('some error in code')

finally:

 print('End of code')

Output:

some error in code

End of code

Raising exception using raise statement:

Example:

```
def addition(a,b):  
    if a + b == 0:  
        raise Exception  
    else:  
        return a + b  
  
try:  
    addition(0,0)  
except:  
    print('the values are zero')
```

Output:

the values are zero