

Exception Handling in Hindi - Python in Hindi

 hindilearn.in/tut/python/exception-handling-in-hindi

Python - Exception Handling

What is Exception ?

Exceptions ये Python में Errors के प्रकार होते हैं | Exception के अलग-अलग प्रकार पड़ते हैं | हर अलग-अलग error के लिए अलग-अलग exception होता है |

जब Python के program पर error occurred होता है तब कोई ना कोई exception raise होता है |

जब program पर किसी statement के लिए exception raise होता है तो उस statement के आगे की script execution की वजह से execute नहीं होती है |

इन scripts को executes करने के लिए error/exception को handle करना पड़ता है |

For Example,

Example पर तीन variables को defined करके print किया गया है | लेकिन जहा पर print statements हैं वहा पर अतिरिक्त undefined 'd' variable को print करने की अनुमति दी गयी है |

'd' variable defined न होने की वजह से exception raise होता है | 'd' के print statement से पहले का print statement execute होता है लेकिन उसके बाद के statements print नहीं हो पाते हैं |

उन अगले statements को print करने के लिए exception को handle करना पड़ता है |

Output :

Example for Exception Handling

Example पर Exception को handle किया गया है |

Output :

```
10
Variable is not defined
20
30
```

try_except Statement(Exception Handling) in Python

Python में error को handle करना हो तो 'try_except' statement का इस्तेमाल करना पड़ता है | Python में जिस code पर error occurred होने की संभावना होती है वो code 'try' block पर आता है और except block पर exception/error को handle करने के लिए कुछ code दिया जाता है |

Syntax for 'try_except' Statement

```
try:
    try_block_code
except Exception1:
    exception handling code 1
except Exception2:
    exception handling code 2
```

Syntax में देखे तो एक try block के लिए एक या एक से अधिक except clauses लिए जा सकते हैं |

Example for 'try_except' Statement

Example में try block पर जो code दिया है वो 'ZeroDivisionError' exception raise करता है इसीलिए ZeroDivisionError का ही except clause execute होगा |

Source Code :

```
try:
    a = 10
    a/0
except NameError:
    print("'a' is not defined")
except ZeroDivisionError:
    print("Divided By Zero Error")
```

Output :

Divided By Zero Error

'try_except' with 'else' Statement(Exception Handling) in Python

Syntax for 'try_except' Statement

```
try:
    try_block_code
except Exception1:
    exception handling code 1
except Exception2,Exception3:
    exception handling code 2
else:
    else_block_code
```

Syntax में देखे तो एक try block के लिए एक या एक से अधिक except clauses लिए जा सकते हैं |

जब try block code में दिया हुआ code कोई भी exception raise नहीं करता है तो else block code execute होता है |

Example में try block code कोई exception raised नहीं करता है इसीलिए else block code execute होगा |

Source Code :

```
try:
    a = 10
    a/4
except NameError:
    print("'a' is not defined")
except ZeroDivisionError:
    print("Divided By Zero Error")
else:
    print("No Exception raised")
```

Output :

No Exception raised

except clause with More than one Exception

Syntax :

```
try:
    try_block_code
except Exception1:
    exception handling code 1
except (Exception2, Exception3):
    exception handling code 2
```

Syntax में देखे तो एक try block के लिए एक या एक से अधिक except clauses लिए जा सकते हैं और हर except clause में एक या एक से अधिक exceptions दिए जा सकते हैं | लेकिन उन multiple exceptions को parenthesis(()) में close करना पड़ता है |

Example : Output :

Something is wrong

try_except_finally Block(Exception Handling) in Python

Syntax :

```
try:  
    try_block_code  
except Exception(s):  
    exception_handling_code  
finally:  
    always_executed_code
```

try block का code जब कोई exception raised करता है और उसे handle नहीं किया जाता है तो finally block का code पहले execute होता है |

try block का code जब कोई exception raised करता है और उसे handle किया जाता है तो except block का code पहले execute होता है और बाद में finally block का code execute होता है |

अगर exception raised होता है या नहीं होता है finally block का code हमेशा execute होता है |

Source Code : Output :

```
Value must be a Number  
Alway executed
```

Raising Exceptions

Exception user द्वारा raise करने 'raise statement' का इस्तेमाल किया जाता है |

raise Statement program में कोई भी exception raised करने के लिए मजबूर कर देता है |

Syntax :

```
raise exception('message') #argument is optional
```

Source Code :

```
| raise TypeError("HI")
```

Output :

```
raise TypeError("HI")  
TypeError: HI
```

Another Example for Raising Exception

Source Code :

```
try:
    raise ValueError('ValueError raised forcely')
except Exception as e:
    print ('Error :',e)
```

Output :

Error : ValueError raised forcely