

Python Exception Handling

Unlock the world of coding

Pankaj Chouhan

www.codeswithpankaj.com

What are Exceptions ?

Exception handling is an essential concept in Python that allows developers to manage errors gracefully without crashing their programs.

An exception is an error that occurs during the execution of a program, disrupting the normal flow of the program.

For example :

```
print(10 / 0) # ZeroDivisionError: division by zero
```

Instead of letting the program crash, we can handle this error using Python's built-in exception handling mechanism.

Common Python Exceptions

- **ZeroDivisionError** : Division by zero.
- **TypeError** : Invalid operation between different data types.
- **ValueError** : Incorrect value given to a function.
- **IndexError** : Accessing an invalid index in a list.
- **KeyError** : Accessing a non-existent dictionary key.
- **FileNotFoundError** : Trying to open a file that doesn't exist.

Unlock the world of coding

Basic Exception Handling with try and except

The try block is used to test a piece of code that may raise an exception. If an exception occurs, it is caught by the except block.

```
print(10 / 0)  
# ZeroDivisionError : division by zero
```



```
try:  
    x = 10 / 0 # This will raise ZeroDivisionError  
except ZeroDivisionError:  
    print("Error: You cannot divide by zero!")
```

Handling a Single Exception

Unlock the world of coding

```
print(10 / 0)
# ZeroDivisionError
: division by zero
```



```
try:
    num = int(input("Enter a number: "))
# User might enter a non-numeric
value
    result = 10 / num # Might cause
ZeroDivisionError
except ZeroDivisionError:
    print("Error: Division by zero is not
allowed.")
except ValueError:
    print("Error: Invalid input! Please
enter a numeric value.")
```

Unlock the world of coding

Handling Multiple Exceptions

Using else and finally

- else Block: Executes if no exception occurs.
- finally Block: Always executes, whether an exception occurs or not.

```
print(10 / 0)  
# ZeroDivisionError  
: division by zero
```

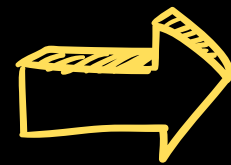


```
try:  
    num = int(input("Enter a number: "))  
    result = 10 / num  
except ZeroDivisionError:  
    print("Error: Cannot divide by zero!")  
except ValueError:  
    print("Error: Enter a valid number!")  
else:  
    print(f"Result: {result}") # Runs if no  
exception occurs  
finally:  
    print("Execution completed.") #  
Always runs
```

Handling Multiple Exceptions with a Single except Block

- Instead of writing multiple except blocks, we can use a tuple to catch multiple exceptions in a single block.

```
print(10 / 0)  
# ZeroDivisionError  
: division by zero
```



```
try:  
    num = int(input("Enter a number: "))  
    result = 10 / num  
except (ZeroDivisionError, ValueError)  
as e:  
    print(f"Error occurred: {e}")
```

Unlock the world of coding