

Understanding Errors and Exceptions in Python

Python developers often encounter **errors** and **exceptions** while coding. Learning to recognize, debug, and handle these is key to improving your programming skills.

1. Errors

Errors occur when there is a fundamental issue with the code, preventing it from running. These are typically **syntax issues** caused by human mistakes.

- **Syntax Errors:**

- Occur due to incorrect syntax in the code.
- Examples:

```
# Missing colon
if True
    print("Hello")
```

```
# Output: SyntaxError: invalid syntax
```

- Common syntax errors:
 - Missing colons (:) in control structures.
 - Mismatched parentheses or brackets.
 - Improper indentation.
- IDEs and editors like Visual Studio Code often highlight syntax errors, making them easier to identify.

2. Exceptions

Exceptions occur during the execution of a program and disrupt its normal flow. Unlike syntax errors, exceptions indicate a problem that can be **handled programmatically**.

- **Common Exception Types:**

- **ZeroDivisionError** : Division by zero.

```
print(5 / 0)
# Output: ZeroDivisionError: division by zero
```

- **FileNotFoundError** : Trying to access a file that doesn't exist.

```
open("non_existent_file.txt")  
# Output: FileNotFoundError: [Errno 2] No such file or directory
```

- **ValueError** : Invalid value.

```
int("not_a_number")  
# Output: ValueError: invalid literal for int() with base 10
```

- **IndexError** : Accessing an invalid index in a list.

```
my_list = [1, 2, 3]  
print(my_list[5])  
# Output: IndexError: list index out of range
```

3. Handling Exceptions

You can handle exceptions to prevent your program from crashing.

- Using `try` and `except` :

```
try:  
    result = 5 / 0  
except ZeroDivisionError:  
    print("You cannot divide by zero!")  
# Output: You cannot divide by zero!
```

- Handling Multiple Exceptions:

```
try:  
    number = int("abc")  
except ValueError:  
    print("Invalid input!")  
except ZeroDivisionError:  
    print("Division by zero is not allowed!")
```

- Using a Generic Exception:

```
try:  
    print(5 / 0)  
except Exception as e:
```

```
print(f"An error occurred: {e}")  
# Output: An error occurred: division by zero
```

Summary

Feature	Errors	Exceptions
Definition	Syntax problems in the code.	Issues during code execution.
Examples	Missing colons, indentation errors.	Division by zero, file not found.
How to Handle	Fix code manually.	Use <code>try - except</code> blocks.

Mastering error and exception handling makes your programs more robust and user-friendly.