Built-in Errors

How to read an Error?

Last line = the actual exception → e.g. IndentationError: expected an indented block after ‘if’ statement on line 1

Lines above = where it happened (the traceback). Start from the bottom, then look at the file/line just above the last line.

```
ERROR!
Traceback (most recent call last):
  File "<main.py>", line 2
    print("True")
    ^^^^^
IndentationError: expected an indented block after 'if' statement on
    line 1


=== Code Exited With Errors ===
```

Most Common Errors

SyntaxError

- **What it means**: Python couldn't parse your code.
- **Common triggers**: missing **)**, **]**, **:**, stray commas, typos in keywords.
- **Quick fix**: jump to the line shown and check recent edits.

```
if n % 2 == 0    # ← missing colon
    print("even")
```

```
ERROR!
Traceback (most recent call last):
  File "<main.py>", line 1
    if n % 2 == 0    # ← missing colon
                     ^^^^^^^^^^^^^^^^^^
SyntaxError: expected ':'
```

IndentationError

- **What it means**: indentation is wrong or mixed tabs/spaces.
- **Fix**: use 4 spaces per indent, never tabs; let your editor fix indentation.

```
if 2 > 3:
print("True") # ← No Indentation
```

```
ERROR!
Traceback (most recent call last):
  File "<main.py>", line 2
    print("True")
    ^^^^^
IndentationError: expected an indented block after 'if' statement on
    line 1
```

NameError

- **What it means**: you used a variable/function name that doesn't exist yet.
- **Triggers**: typos; using a variable before assigning it.
- **Fix**: check spelling and order of definitions.

```
print(usernme) # ← NameError
```

```
ERROR!
Traceback (most recent call last):
  File "<main.py>", line 1, in <module>
NameError: name 'usernme' is not defined
```

TypeError

- **What it means**: you used the right idea but wrong type or wrong number of arguments.
- **Triggers**: "2" + 2, calling a non-callable, wrong param count.
- **Fix**: convert or adjust arguments.

```
print("2" + 2)
```

```
ERROR!
Traceback (most recent call last):
  File "<main.py>", line 1, in <module>
TypeError: can only concatenate str (not "int") to str
```

ValueError

- **What it means**: type is fine, but the value isn't.
- **Triggers**: int("abc")
- **Fix**: validate or sanitize input first; handle with try/except.

```
name = int(input("Enter a number: "))
```

```
Enter a number: five
ERROR!
Traceback (most recent call last):
  File "<main.py>", line 1, in <module>
ValueError: invalid literal for int() with base 10: 'five'
```

AttributeError

- **What it means**: the object doesn't have that attribute/method.
- **Triggers**: None.some_method(), typos like appendd.
- **Fix**: print the object/type; check docs or dir(obj).

```
[1,2].strip() # ← AttributeError
```

```
ERROR!
Traceback (most recent call last):
  File "<main.py>", line 1, in <module>
AttributeError: 'list' object has no attribute 'strip'
```

IndexError

- **What it means**: list/tuple index is out of range.
- **Fix**: check length before indexing; use safe loops.

```
nums = [10,20]
print(nums[4])
```

```
ERROR!
Traceback (most recent call last):
  File "<main.py>", line 2, in <module>
IndexError: list index out of range
```

KeyError

- **What it means**: dict key doesn't exist.
- **Fix**: use **in**, **.get()**, or **dict.setdefault**.

```
fruit = {"name": "apple", "color": "red"}
print(fruit["price"])
```

```
ERROR!
Traceback (most recent call last):
  File "<main.py>", line 2, in <module>
KeyError: 'price'
```

ZeroDivisionError

- **What it means**: divisor is zero.
- **Fix**: guard against zero; show a friendly message or special-case.

```
print(10/0)
```

```
ERROR!
Traceback (most recent call last):
  File "<main.py>", line 1, in <module>
ZeroDivisionError: division by zero
```

FileNotFoundError

- **What it means**: the file/path doesn't exist.
- **Fix**: check the path, working directory, and spelling; create file if needed

```
open("txt")
```

```
ERROR!
Traceback (most recent call last):
  File "<main.py>", line 1, in <module>
FileNotFoundError: [Errno 2] No such file or directory: 'txt'
```

ImportError / ModuleNotFoundError

- **What it means**: Python couldn't import something.
- **Triggers**: package not installed, wrong module name, wrong import path.
- **Fix**: pip install package, check spelling, or use the right import.

```
import wtfit                    # ← ModuleNotFoundError
from random import abc          # ← ImportError
```

```
ERROR!
Traceback (most recent call last):
  File "<main.py>", line 1, in <module>
ModuleNotFoundError: No module named 'wtfit'
```

```
ERROR!
Traceback (most recent call last):
  File "<main.py>", line 1, in <module>
ImportError: cannot import name 'abc' from 'random' (/usr/local/lib
    /python3.12/random.py)
```

AssertionError

What it means: Raised when an assert statement fails (the condition is False).

Fix: Make sure the condition inside assert is correct, or replace with explicit error handling if it's user input (since assert is mainly for debugging, not for validating input).

```
x = -5
assert x >= 0, "x must be non-negative"
```

```
ERROR!
Traceback (most recent call last):
  File "<main.py>", line 2, in <module>
AssertionError: x must be non-negative
```

Other Errors

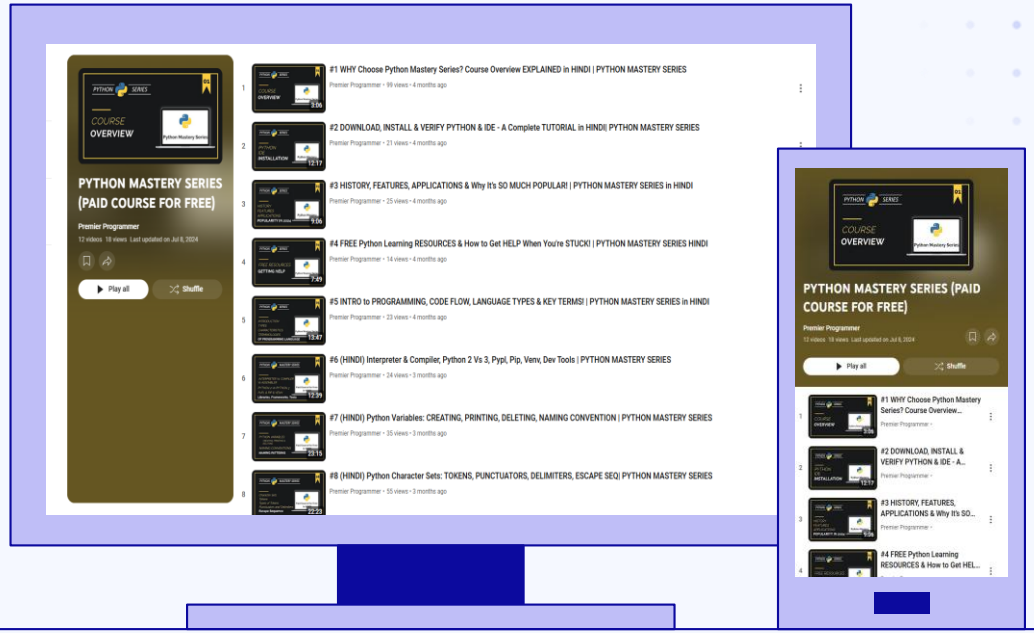
ArithmeticError	AttributeError	Exception	EOFError	FloatingPointError
GeneratorExit	KeyboardInterrupt	LookupError	MemoryError	NotImplementedError
OSError	OverflowError	ReferenceError	RuntimeError	StopIteration
TabError	SystemError	SystemExit	UnboundLocalError	UnicodeError
UnicodeEncodeError	UnicodeDecodeError	UnicodeTranslateError		

Mega Set

Download Link in **Description** and **Pinned Comment**

WATCH

Level up your coding with each episode in this focused Python series.



Next Video!

Mega Set Solution