



Python Try Except

[< Previous](#)[Next >](#)

The `try` block lets you test a block of code for errors.

The `except` block lets you handle the error.

The `else` block lets you execute code when there is no error.

The `finally` block lets you execute code, regardless of the result of the try- and except blocks.

Exception Handling

When an error occurs, or exception as we call it, Python will normally stop and generate an error message.

These exceptions can be handled using the `try` statement:

Example

[Get your own Python Server](#)

The `try` block will generate an exception, because `x` is not defined:

```
try:  
    print(x)
```

```
except:  
    print("An exception occurred")
```

[Try it Yourself »](#)

Since the try block raises an error, the except block will be executed.

Without the try block, the program will crash and raise an error:

Example

This statement will raise an error, because `x` is not defined:

```
print(x)
```

[Try it Yourself »](#)

Many Exceptions

You can define as many exception blocks as you want, e.g. if you want to execute a special block of code for a special kind of error:

Example

Print one message if the try block raises a `NameError` and another for other errors:

```
try:  
    print(x)  
except NameError:  
    print("Variable x is not defined")  
except:  
    print("Something else went wrong")
```

[Try it Yourself »](#)

Else

You can use the `else` keyword to define a block of code to be executed if no errors were raised:

Example

In this example, the `try` block does not generate any error:

```
try:
    print("Hello")
except:
    print("Something went wrong")
else:
    print("Nothing went wrong")
```

[Try it Yourself »](#)

Finally

The `finally` block, if specified, will be executed regardless if the try block raises an error or not.

Example

```
try:
    print(x)
except:
    print("Something went wrong")
```

```
finally:  
    print("The 'try except' is finished")
```

[Try it Yourself »](#)

This can be useful to close objects and clean up resources:

Example

Try to open and write to a file that is not writable:

```
try:  
    f = open("demofile.txt")  
    try:  
        f.write("Lorum Ipsum")  
    except:  
        print("Something went wrong when writing to the file")  
    finally:  
        f.close()  
except:  
    print("Something went wrong when opening the file")
```

[Try it Yourself »](#)

The program can continue, without leaving the file object open.

Raise an exception

As a Python developer you can choose to throw an exception if a condition occurs.

To throw (or raise) an exception, use the `raise` keyword.

Example

Raise an error and stop the program if x is lower than 0:

```
x = -1

if x < 0:
    raise Exception("Sorry, no numbers below zero")
```

[Try it Yourself »](#)

The `raise` keyword is used to raise an exception.

You can define what kind of error to raise, and the text to print to the user.

Example

Raise a `TypeError` if x is not an integer:

```
x = "hello"


if not type(x) is int:
    raise TypeError("Only integers are allowed")
```

[Try it Yourself »](#)[◀ Previous](#)[Next ▶](#)



COLOR PICKER





Master Web Dev
with **Live**
Instructors
W3Schools
Bootcamps
6 Months Fully
Online - Only \$2,995

[Learn more](#)

[Report Error](#)

[Spaces](#)

[Upgrade](#)

[Newsletter](#)

[Get Certified](#)

Top Tutorials

[HTML Tutorial](#)
[CSS Tutorial](#)
[JavaScript Tutorial](#)
[How To Tutorial](#)
[SQL Tutorial](#)
[Python Tutorial](#)
[W3.CSS Tutorial](#)
[Bootstrap Tutorial](#)
[PHP Tutorial](#)
[Java Tutorial](#)
[C++ Tutorial](#)
[jQuery Tutorial](#)

Top References

- HTML Reference
- CSS Reference
- JavaScript Reference
- SQL Reference
- Python Reference
- W3.CSS Reference
- Bootstrap Reference
- PHP Reference
- HTML Colors
- Java Reference
- Angular Reference
- jQuery Reference

Top Examples

- HTML Examples
- CSS Examples
- JavaScript Examples
- How To Examples
- SQL Examples
- Python Examples
- W3.CSS Examples
- Bootstrap Examples
- PHP Examples
- Java Examples
- XML Examples
- jQuery Examples

Get Certified

- HTML Certificate
- CSS Certificate
- JavaScript Certificate
- Front End Certificate
- SQL Certificate
- Python Certificate
- PHP Certificate
- jQuery Certificate
- Java Certificate
- C++ Certificate
- C# Certificate
- XML Certificate

[FORUM](#) | [ABOUT](#)

W3Schools is optimized for learning and training. Examples might be simplified to improve reading and learning. Tutorials, references, and examples are constantly reviewed to avoid errors, but we cannot warrant full correctness of all content. While using W3Schools, you agree to have read and accepted our terms of use, cookie and privacy policy.

W3Schools is Powered by W3.CSS.

