

Python Try Except

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

The `except` block lets you handle the 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:

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

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

Output: An exception occurred

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:

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

```
print(x)
```

Output: NameError: name 'x' is not defined

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:

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")
```

Output: Variable x is not defined

Else

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

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

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

Output:

Hello

Nothing went wrong

Finally

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

```
try:
    print(x)
except:
    print("Something went wrong")
finally:
    print("The 'try except' is finished")
```

Output:

```
Something went wrong
The 'try except' is finished
```

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

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

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

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

Output: Something went wrong when writing to the file