

# Python Try Except

[https://www.w3schools.com/python/python\\_try\\_except.asp](https://www.w3schools.com/python/python_try_except.asp)

## Examples

- Блок **try** позволяет проверить блок кода на наличие ошибок.
- Блок **except** позволяет обработать ошибку.
- Блок **else** позволяет выполнять код при отсутствии ошибок.
- Блок **finally** позволяет выполнять код независимо от результата блоков try- и exclude.

## Exception Handling - Обработка исключений

Когда возникает ошибка или исключение, как мы это называем, Python обычно останавливается и генерирует сообщение об ошибке.

Эти исключения могут быть обработаны с помощью `try` оператора:

Блок `try` сгенерирует исключение, потому что `x` не определено:

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

Поскольку блок `try` вызывает ошибку, **будет выполнен блок `except`**.

Без блока `try` программа вылетит и выдаст ошибку:

## Many Exceptions

Вы можете определить столько блоков исключений, сколько хотите, например, если вы хотите выполнить специальный блок кода для особого вида ошибки:

Выведите одно сообщение, если блок `try` вызывает а, `NameError` и другое для других ошибок:

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

## 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

```
x = 5
try:
    print(x + 2)
except NameError:
    print("Variable x is not defined")
else:
    print("Something else went wrong")
```

Output:  
7  
Something else went wrong

## Finally

Блок **finally**, если он указан, будет выполнен независимо от того, вызовет ли блок **try** ошибку или нет.

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

Output:  
Something went wrong  
The 'try except' is finished

Это может быть полезно для закрытия объектов и очистки ресурсов:

```
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")
```

Output:  
Something went wrong when writing to the file

## Raise an exception - Создать исключение

Как разработчик Python, вы можете выбрать исключение, если возникнет условие.

Чтобы сгенерировать (или вызвать) исключение, используйте `raise` ключевое слово.

Выдать ошибку и остановить программу, если x меньше 0:

```
x = -1
if x < 0:
    raise Exception("Sorry, no numbers below zero")
```

Ключевое `raise` слово используется для создания исключения.

Вы можете определить, какую ошибку вызывать и какой текст выводить пользователю.

Поднимите `TypeError`, если x не является целым числом:

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