

# Исключения в Python

---

*Берленко Т.А. СПбГЭТУ “ЛЭТИ”, ФКТИ, МОЭВМ*

# Ошибки и исключения

- Ошибки и исключения - это особый класс объектов в языке Python.
- **Синтаксические** ошибки возникают в процессе компиляции программы на языке Python.
- Остальные ошибки (или исключения) возникают во время исполнения программы.
- Ошибки могут возникать в случае некорректного программного кода, а также генерироваться программистом.

## Ошибки и исключения

- Пример синтаксической ошибки:

```
data = '1'
```

```
    print(data)
```

```
# print(data)
```

```
# ^
```

```
# IndentationError: unexpected indent
```

## Ошибки и исключения

➤ Пример исключения ValueError:

```
data = '1q'  
print(int(data))
```

# Мы поймаем исключение ValueError:

# ValueError: invalid literal for int() with base 10: '1q'

# Ошибки и исключения

У любой ошибки есть:

- Тип (класс), например: `TypeError`, `ValueError`, ...
- Сообщение, например:  
**“invalid literal for int() with base 10: '1q'”**
- Состояние стека вызовов на момент ошибки.

# Конструкция try-except

- Конструкция try-except нужна для того, чтобы перехватить и обработать исключительные ситуации.
- Синтаксис:

try:

    <Инструкция>

except <Тип\_Исключения>:

    <Обработка\_Исключения>

# Конструкция try-except

Пример:

```
data = input()
```

```
try:
```

```
    data = int(data)
```

```
except ValueError:
```

```
    print('Это не число!')
```

## Конструкция try-except as

- В конструкции try-except мы также можем поймать объект ошибки, для этого используется следующий синтаксис:

```
try:  
    <Инструкция>  
except <Тип_Исключения> as <Имя_Объекта_Исключения>:  
    <Обработка_Объекта_Исключения>
```



## Конструкция try-except as

```
def func(arr):  
    try:  
        print(min(arr))  
    except TypeError as e:  
        print(e)
```

```
func([1, 2, 3, 'f'])
```

# Иерархия исключений

- Все исключения в языке Python участвуют в иерархии наследования
- Базовый класс для всех исключений - `BaseException`, но программистам рекомендуется использовать `Exception` для создания собственных исключений.
- Поскольку при обнаружении исключительной ситуации в `try-except` используется функция **`isinstance()`**, мы можем отлавливать как объект указанного исключения, так и объект классов-наследников этого исключения.

# Конструкция try-except-else-finally

**try:**

    <Инструкция>

**except** <Тип\_Исключения>:

    <Обработка\_Исключения>

**else:**

    # код для обработки случая,

    # когда в try-блоке не было поймано исключение <Тип\_Исключения>

**finally:**

    # код, который нужно выполнить при любом исходе

# Инструкция raise

- Мы можем сгенерировать (или бросить) исключение с помощью инструкции raise.

- Синтаксис:

`raise <Создание объекта исключения>`

- Пример:

`raise TypeError('Тип переменной указан неверно!')`