



Исключения

Исключения – это ошибки, которые выбрасываются при неправильной работе программы и останавливают её выполнение, если они не обработаны.

Синтаксис отлова ошибок:

```
try:
    *код, который может
    вызвать ту или иную ошибку*
    except *ошибка*:
        *код который выполнится
        в случае возникновения
        ошибки*
else:
    *код который выполнится
    только в случае если в try
    ничего не сломалось*
finally:
    *код который выполнится по
    любому*
```

Блоки finally и else, являются не обязательными, но могут быть использованы для вашего удобства.

Fullstack разработчик на Python

Статические методы

Статические методы – это методы которые могут вызываться напрямую через класс и не зависят от состояния объекта.

Синтаксис статического метода:

```
@staticmethod
def method(*args, **kwargs):
    *код статического метода*
```

Декораторы класса

@property – декоратор для представления метода как свойства (используется как геттер)

@.setter – декоратор дополняющий @property для установки новых значений в данное свойство.

Пример методов с декораторами:

```
@property
def a(self):
    return self._a
```

```
@a.setter
def a(self, value):
    self._a = value
```

@classmethod – декоратор который удобно использовать при полиморфизме. Вместо самого объекта он передаёт модель класса первым аргументом.

Модуль Типы данных

Пример методов с декораторами:

```
@property
def a(self):
    return self._a
```

```
@a.setter
def a(self, value):
    self._a = value
```

Синтаксис классowego метода:

```
@classmethod
def method(cls, *args, **kwargs):
    cls.originalMethod(<args>)
```