GeekBrains

ООП. Полезные дополнения

Основы Python

Что будет на уроке

1. Статические методы и методы класса.
2. Атрибуты и встроенные методы объектов класса.
3. Пример ООП-программы.
4. Создание собственных исключений.
5. Библиотека psutil.
6. Pip и virtualenv. Особенности использования.
7. Библиотека requests.

Статические методы и методы класса

@staticmethod

Декоратор для метода класса. Такой метод вызывается напрямую через имя класса (статический метод).

@classmethod

Декоратор для метода класса. Такой метод получает класс в качестве первого аргумента.

Атрибуты и встроенные методы объектов классов. Часть 1

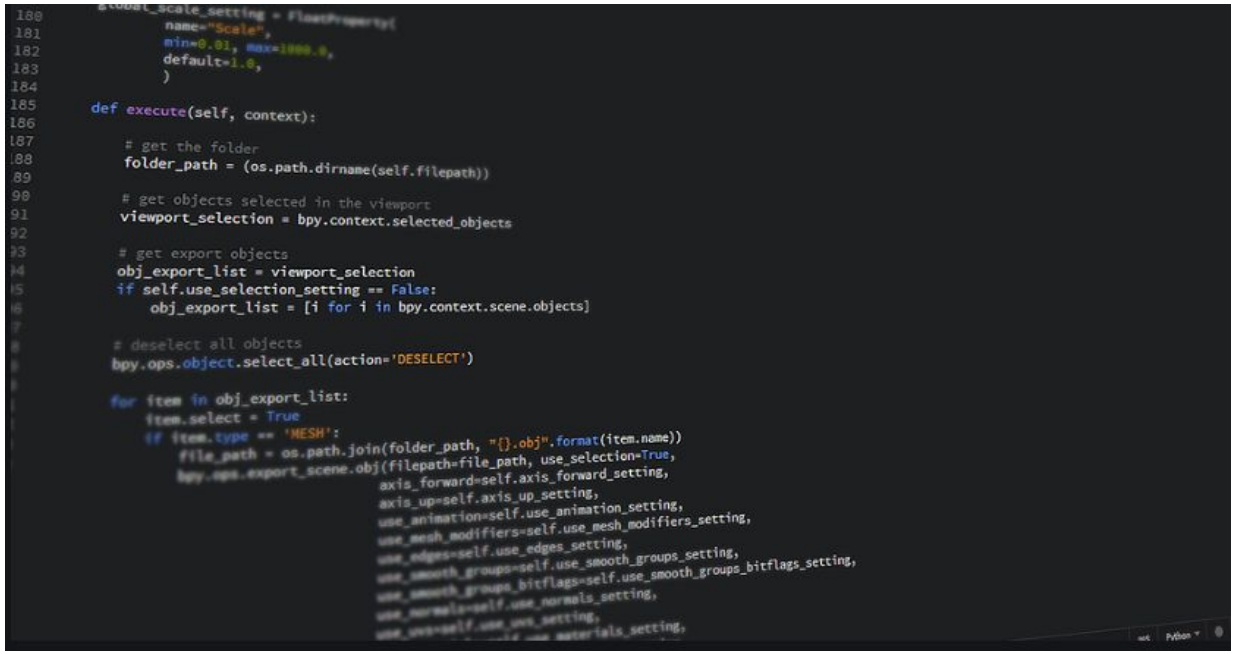
Атрибут	Описание
`__name__`	Имя класса
`__module__`	Имя модуля
`__dict__`	Словарь с атрибутами класса
`__bases__`	Кортеж с базовыми классами
`__doc__`	Строка документации класса
`__class__`	Объект-класс, экземпляром которого является этот инстанс
`__init__`	Конструктор
`__del__`	Деструктор
`__hash__`	Возвращает хеш-значение объекта, равное 32-битному числу

Атрибуты и встроенные методы объектов классов. Часть 2

Атрибут	Описание
`__getattr__`	Возвращает атрибут, недоступный обычным способом
`__setattr__`	Присваивает значение атрибуту
`__delattr__`	Удаляет атрибут
`__call__`	Выполняется при вызове экземпляра класса
`__str__`	Строковое представление объекта
`__repr__`	Формальное строковое представление объекта
`__getitem__`	Получение элемента по индексу или ключу
`__setitem__`	Присваивание элемента с данным ключом или индексом
`__delitem__`	Удаление элемента с данным ключом или индексом

ООП-программа

1. Сформулировать задачу.
2. Определить объекты предметной области, участвующие в решении задачи.
3. Выделить классы, на основе которых генерируются объекты. При необходимости определить базовые классы и классы-потомки.
4. Установить основные атрибуты и методы объектов.
5. Создать классы, их атрибуты и методы.
6. Создать объекты классов.
7. Выполнить итоговое решение задачи, организовав взаимодействие объектов.



Создание собственных исключений

В Python существует возможность создания собственных классов-исключений — потомков класса Exception.

Exception
ZeroDivisionError
IndexError
KeyError
FileExistsError
FileNotFoundError
IndentationError
TypeError
ValueError

Pip. Особенности использования

Это популярная система управления пакетами. Она участвует в установке и управлении программными пакетами, реализованными с помощью Python.

Команда	Описание
pip help	Получить подсказку о доступных командах
pip install package_name	Установить пакет
pip uninstall package_name	Удалить пакет
pip list	Получить список установленных пакетов
pip search package_name	Найти пакет по имени
pip install -U package_name	Обновить указанный пакет
pip show package_name	Получить информацию об установленном пакете

Virtualenv. Особенности использования

Под виртуальной средой понимают директорию, содержащую необходимые для работы приложения пакеты. С их помощью выполняется изолированный запуск приложения.

Установка virtualenv:

```
pip install virtualenv
```

Создание виртуальной среды:

```
virtualenv my_proj
```

Активация виртуальной среды:

```
my_proj\Scripts\activate - Windows
source my_proj/venv/bin/activate - Linux, MacOS
```

Деактивация виртуальной среды:

```
my_proj\Scripts\deactivate - Windows
deactivate - Linux, MacOS
```

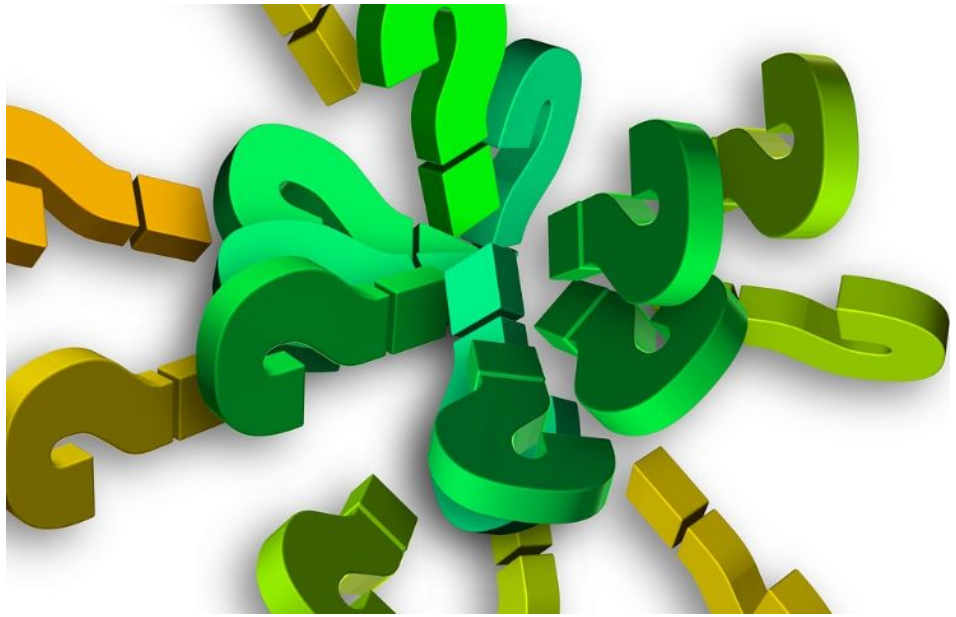
Библиотека psutil

Позволяет получить информацию о параметрах процессора, памяти, дисков.

Это отличная библиотека для управления системой и ресурсами.

Библиотека requests

Сторонний инструмент для выполнения запросов и обработки ответов. Одно из ключевых звеньев для парсинга веб-страниц.



Итоги урока

1. Познакомились с полезными дополнениями механизма ООП в Python: статическим методом и методом класса.
2. Рассмотрели атрибуты и встроенные методы объектов класса.
3. Научились пошагово создавать ООП-программы и собственные исключения.
4. Узнали о некоторых трюках для создания лаконичного и стильного кода.
5. Познакомились с библиотеками: psutil, pip, virtualenv, requests.