

# Python Essential

ООП - Классы, атрибуты, методы, конструктор

# Python Essential

После урока обязательно



Повторите этот урок в видео формате на [ITVDN.com](http://itvdn.com)



Проверьте как Вы усвоили данный материал на [TestProvider.com](http://testprovider.com)

# ООП - Классы, атрибуты, методы, конструктор

# Python Essential

## Парадигмы программирования

**Парадигма программирования** – это совокупность идей и понятий, определяющих стиль написания компьютерных программ, подход к программированию.

Python поддерживает разные парадигмы программирования

- императивное программирование
- процедурное программирование
- структурное программирование
- объектно-ориентированное программирование
- функциональное программирование



# Python Essential

## Объектно-ориентированное программирование

**Объектно-ориентированное программирование (ООП)** – парадигма программирования, в которой основными концепциями являются понятия объектов и классов.

**Класс** является моделью ещё не существующей сущности (объекта). Он является составным типом данным, включающим в себя поля и методы.

**Объект** – это экземпляр класса.

Основные принципы ООП:

- Абстракция
- Инкапсуляция
- Полиморфизм
- Наследование



# Python Essential

## ООП в Python

В Python **всё** является объектами – экземплярами каких-либо классов, даже сами классы, которые являются объектами – экземплярами метаклассов. Главным метаклассом является класс `type`, который является абстракцией понятия типа данных.



**Всё есть объект**

# Python Essential

## Классы в Python

- В терминологии Python члены класса называются **атрибутами**. Эти атрибуты могут быть как переменными, так и функциями.
- Классы создаются при помощи ключевого слова **class**.
- Классы как объекты поддерживают два вида операций: обращение к атрибутам классов и создание (*инстанцирование*) объектов – **экземпляров класса** (instance objects).
- Обращение к атрибутам какого-либо класса или объекта производится путём указания имени объекта и имени атрибута через точку.
- Для создания экземпляров класса используется синтаксис вызова функции.



🔥 Класс! 468

# Python Essential

## Экземпляры классов в Python

- Единственная доступная операция для объектов-экземпляров – это доступ к их атрибутам.
- Атрибуты *объектов-экземпляров* делятся на два типа: **атрибуты-данные** и **методы**.
- Атрибуты-данные аналогичны *полям* в терминологии большинства широко распространённых языков программирования.
- Атрибуты-данные не нужно описывать: как и переменные, они создаются в момент первого присваивания. Как правило, их создают в методе-конструкторе `__init__`.
- Метод – это функция, принадлежащая объекту. Все атрибуты класса, являющиеся функциями, описывают соответствующие методы его экземпляров, однако они не являются одним и тем же.
- Особенностью методов является то, что в качестве первого аргумента им *передаётся данный экземпляр класса*. Таким образом, если `obj` – экземпляр класса `MyClass`, вызов метода `obj.method()` эквивалентен вызову функции `MyClass.method(obj)`.



Первый аргумент метода, который соответствует текущему экземпляру, принято называть *self*.



# Python Essential

## Разница между атрибутами класса и атрибутами-данными

**Атрибуты класса являются общими для самого класса и всех его экземпляров. Их изменение отображается на все соответствующие объекты. Атрибуты-данные принадлежат конкретному экземпляру и их изменение никак не влияет на соответствующие атрибуты других экземпляров данного класса.** Таким образом, атрибуты класса, которые не являются функциями, примерно соответствуют статическим полям в других языках программирования, а атрибуты-данные – обычным полям.

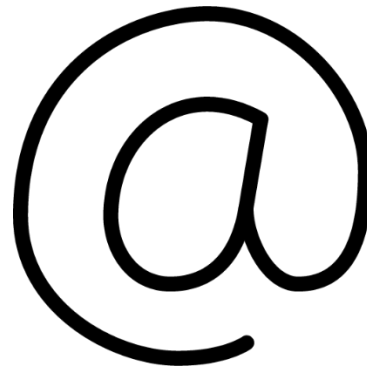


**Следует понимать разницу между атрибутами класса и атрибутами-данными**

# Python Essential

## Статические методы и методы класса

- **Декоратор** – это специальная функция, которая изменяет поведение функции или класса. Для применения декоратора следует перед соответствующим объявлением указать символ @, имя необходимого декоратора и список его аргументов в круглых скобках. Если передача параметров декоратору не требуется, скобки не указываются.
- Для создания **статических методов** используется декоратор `staticmethod`.
- Для создания **методов класса** используется декоратор `classmethod`.



Методы класса похожи на обычные методы, но относятся к самому классу как объекту – экземпляру метакласса (в отличие от обычных методов, которые принадлежат объектам – экземплярам классов, и статических методов, которые относятся к самому классу и всем его экземплярам и не принадлежат никакому объекту – экземпляру). Их первый аргумент принято называть *cls*.

# Python Essential

## Атрибуты в Python

- Все атрибуты по умолчанию являются **публичными**.
- Атрибуты, имена которых начинаются с *одного знака подчёркивания* (`_`) говорят программисту о том, что они относятся ко внутренней реализации класса и не должны использоваться извне, однако *никак не защищены*.
- Атрибуты, имена которых начинаются, но не заканчиваются, *двумя символами подчёркивания*, считаются **приватными**. К ним применяется механизм «**name mangling**». Он не предполагает защиты данных от изменения извне, так как к ним всё равно можно обратиться, зная имя класса и то, как Python изменяет имена приватных атрибутов, однако *позволяет защитить их от случайного переопределения в классах-потомках*.



# Python Essential

## Специальные атрибуты и методы

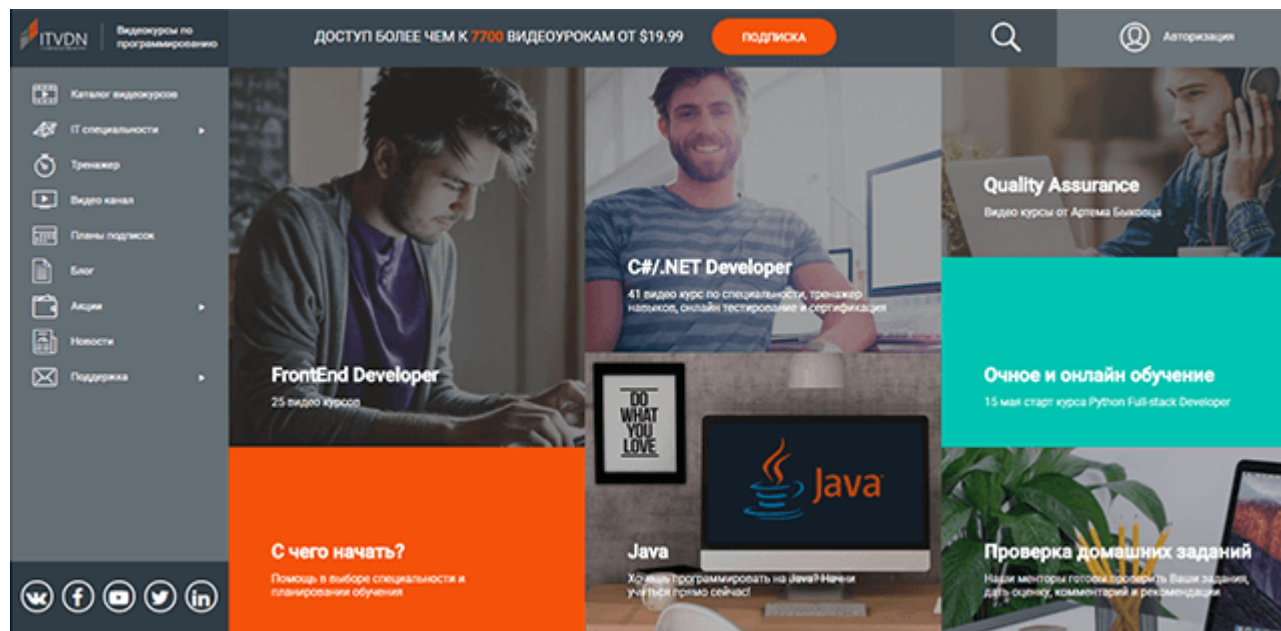
- Атрибуты, имена которых начинаются и заканчиваются *двумя знаками подчёркивания*, являются внутренними для Python и задают особые свойства объектов (примеры: `__doc__`, `__class__`).
- Среди таких атрибутов есть методы. В документации Python подобные методы называются **методами со специальными именами**, однако в сообществе Python-разработчиков очень распространено название «**магические методы**». Также, встречается и название «**специальные методы**». Они задают особое поведение объектов и позволяют переопределять поведение встроенных функций и операторов для экземпляров данного класса.
- Наиболее часто используемым из специальных методов является метод `__init__`, который автоматически вызывается после создания экземпляра класса.



**Не следует объявлять свои собственные (нестандартные) атрибуты с именами, которые начинаются и заканчиваются двумя знаками подчёркивания**

# Смотрите наши уроки в видео формате

ITVDN.com



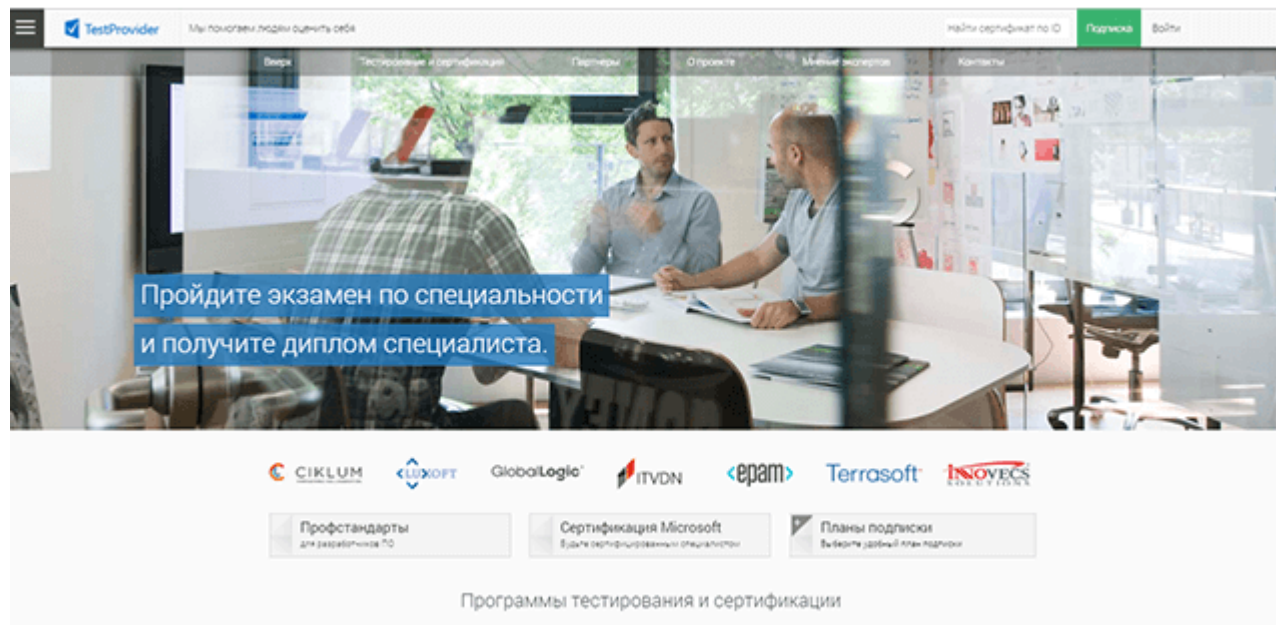
Посмотрите этот урок в видео формате на образовательном портале [ITVDN.com](http://ITVDN.com) для закрепления пройденного материала.

Курсы записаны сертифицированными тренерами, которые работают в учебном центре CyberBionic Systematics и другими высококвалифицированными разработчиками.



# Проверка знаний

TestProvider.com



TestProvider – это online сервис проверки знаний по информационным технологиям. С его помощью Вы можете оценить Ваш уровень и выявить слабые места. Он будет полезен как в процессе изучения технологии, так и для общей оценки знаний IT специалиста.

После каждого урока проходите тестирование для проверки знаний на [TestProvider.com](https://testprovider.com)

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.



# Python Essential

Q&A

# Информационный видеосервис для разработчиков программного обеспечения

