

Python Essential

Модули

Python Essential

После урока обязательно



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

Доступ можно получить через руководство вашего учебного центра



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

Python Essential

Тема

Модули

Модули

Понятие модулей



Модуль — функционально законченный фрагмент программы, оформленный в виде отдельного файла с исходным кодом или поименованной непрерывной её части. Модули позволяют разбивать сложные задачи на более мелкие в соответствии с принципом модульности.

Обычно проектируются таким образом, чтобы предоставлять программистам удобную для многократного использования функциональность (интерфейс) в виде набора функций, классов, констант.

Файл, который содержит исходный код на языке Python, является модулем.

Модули

Импортирование модуля

```
import module_name  
import module_name as new_name
```



Изначальное название импортированного модуля доступно как его глобальная переменная `__name__`. У модуля, запущенного как скрипта, `__name__` равен `"__main__"`.



Модули загружаются только один раз. Все последующие попытки их импортировать лишь возвращают ссылки на уже загруженные модули.

Модули

Импортирование имён из модуля

```
from module import name  
from module import name1, name2  
from module import *
```

```
from module import name as new_name  
from module import name1 as new_name1, name2 as new_name2
```



Если вместо списка имён указать символ *, то импортируются все имена, кроме тех, которые начинаются с символа подчёркивания.



Импортирование всех имён (from module import *) удобно в интерактивной сессии интерпретатора, но нежелательно в коде

Модули

Запуск модуля как скрипта

Глобальная переменная `name` инициализируется значением `"__main__"`, а не `"app"`, так как скрипт запущен, а не импортирован как модуль.

В первом примере
`sys.argv = ['app.py']`

Во втором примере
`sys.argv = ['app.py', '8000']`

Модули

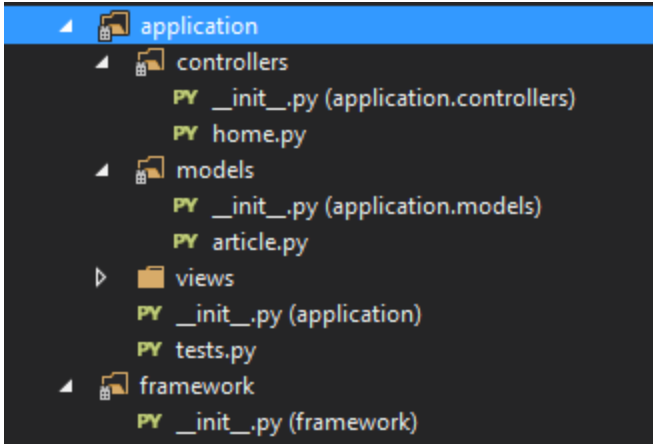
Пути поиска модулей



- При импортировании модулей интерпретатор Python ищет их в директориях и архивах, список которых доступен как для чтения, так и для модификации в виде переменной `path` встроенного модуля `sys`.
- По умолчанию **`sys.path`** состоит из директории с запускаемым скриптом, содержимого переменной окружения `PYTHONPATH` и стандартного расположения модулей, специфичного для конкретной платформы и интерпретатора.
- Для ускорения загрузки модулей Python кеширует байт-код и производит компиляцию модуля только в том случае, если исходный код был изменён. Python 3 сохраняет файлы байт-кода `.pyc` в каталоге `__pycache__`, Python 2 — рядом с файлами исходного кода.

Модули

Пакеты



- Модули могут объединяться в пакеты. Пакеты служат как пространства имён для модулей и способ их структурирования.
- Любой пакет является модулем, но не каждый модуль является пакетом.
- Как правило, модули представляются в виде файлов, а пакеты – каталогов в файловой системе (но не всегда).
- Для того, чтобы каталог был пакетом, в нём должен находиться файл `__init__.py`. Он автоматически выполняется при импортировании соответствующего модуля и может содержать определённые действия для инициализации или быть пустым.

Модули

Импортирование из пакетов

```
import package.module  
import package.subpackage.module  
from package import module  
from package import item  
from package.subpackage import module, item  
from module import *
```



Для того, чтобы можно было импортировать все имена пакета (`from package.subpackage import *`), пакет должен описывать список `__all__`, который содержит имена подпакетов и модулей.



При использовании оператора `from package import item`, `item` может быть пакетом, модулем или любым именем, описанным в пакете. При использовании оператора `import package.item`, `item` должен быть модулем или пакетом.

Модули

Относительное импортирование



Кроме абсолютных, существуют также относительные импорты: точка указывает на текущий пакет, две точки – на родительский. Эти же символы могут быть использованы сразу перед именем пакета или модуля и влиять на то, где интерпретатор будет его искать.

```
from . import name  
from .. import name  
from .package import name  
from ..package import name
```

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.

Все курсы записаны сертифицированными тренерами, которые работают в учебном центре CyberBionic Systematics



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 общей оценки знаний IT специалиста.

После каждого урока проходите тестирование для проверки знаний на TestProvider.com

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.



Python Essential

Q&A

Информационный видеосервис для разработчиков программного обеспечения

