Импортирование

ЛЕКЦИЯ 8

Содержание

- Импортирование модулей
- Импортирование пакетов
- **__init,py__, __name__, __all__**

Основы

Модуль – самая крупная организационная программная единица в языке Python, которая вмещает в себя программный код и данные, готовые для многократного использования.

- В языке Python модули соответствуют файлам с расширением **.py**
- Модули могут импортировать другие модули(файлы) для доступа к именам, которые в них определены.

Импортирование модулей

import – позволяет импортеру получить модуль целиком.

Доступ к именам модуля осуществляется через точку «.»:

```
import math             # модуль математических выражений

print(math.sqrt(4))     # метод извлечения кореня из числа
```

from – позволяет импортеру получить определенное имя из модуля:

```
from math import sqrt

print(sqrt(4))
```

Принципы импорта

1. Отыскивается файл модуля
2. Компилируется в байт-код
3. Запускается программный код модуля, чтобы создать объекты, которые он определяет

Байт – код – адаптированный код программы на **Python**, исполняемый **PVM** (**Python Virtual Machine**)

инструкция **from** и **from ***

Инструкция **from** позволяет импортировать из модуля имя, и использовать его как внутреннее имя текущего модуля:

```
from math import sqrt

print(sqrt(4))
```

Инструкция **from *** позволяет импортировать ВСЕ имена из модуля, и использовать их как внутренние имена текущего модуля:

```
from math import *

print(sqrt(4), sin(pi/2), pi)
```

*Примечание: без необходимости инструкцию from * использовать не рекомендуется, так как может возникнуть конфликт с другими именами*

Повторная загрузка модулей

- Программный код модуля, по умолчанию, вызывается только один раз
- При повторных попытках импортировать модуль будет использоваться объект уже загруженного модуля. Повторная загрузка и запуск программного кода в этом случае не происходит

Функция **reload**() принудительно выполняет повторную загрузку уже загруженного модуля и запускает его программный код. Инструкции присваивания, выполняемые при повторном запуске, будут изменять существующий объект модуля

Функция **reload**() содержится в модуле **importlib**

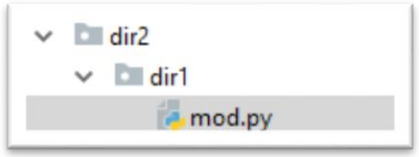
```
import math
from importlib import reload

reload(math)
```

Пакеты

Каталог, внутри которого находятся модули называется **пакетом**, а процедура его импортирования – **импортированием пакета**.

При импортировании пакета, его имя превращается в еще одно пространство имен верхнего уровня, в котором атрибутами являются другие подкаталоги и модули:

```
import dir2.dir1.mod
```



для того чтобы каждый раз не указывать полный путь до модуля можно ввести псевдоним с помощью инструкции **as**

```
import dir2.dir1.mod as md

print(md.a)
```

__name__ и ‘__main__’

У каждого объекта модуля существуют стандартные внутренние поля-атрибуты.

- При импортировании модуля поле __name__ будет содержать строковое имя файла модуля.
- Если же модуль является исполняемым, то полю __name__ присваивается имя ‘__main__’
- Данную особенность можно применять, включая в модули код, который будет вызываться только в случае если модуль является исполняемым

```
import dir2.dir1.mod as md

print('md.__name__ = ', md.__name__)

if __name__ == '__main__':
    print('Это исполняемый модуль!')
```

```
md.__name__ =  dir2.dir1.mod
Это исполняемый модуль!

Process finished with exit code 0
```

__init__.py

Если нужно импортировать пакет, а не конкретный внутренний модуль, следует реализовать внутри пакета файл **__init__.py**

В данном файл будет содержаться код, который исполняется при попытке импортировать пакет

init.py

```
1  import dir2.dir1.mod
```

После этого имя dir можно использовать как пространство имен:

```
import dir2.dir1 as dir1

print('md.__name__ = ', dir1.mod.__name__)
```

__init__.py, __all__ и from*

Чтобы использовать инструкцию **from*** в файле **__init__.py** следует создать список импортируемых модулей и привязать его к зарезервированному имени **__all__ :**

```
__all__ = ["mod"]
```

Теперь при вызове инструкции **from*** для пакета будут импортированы все его внутренние модули:

```
from dir2.dir1 import *

print('md.__name__ = ', mod.__name__)
```