

# Программирование, лекция 17. Прочие модули языка Python. Пакеты

Кафедра ИУ7 МГТУ им. Н. Э. Баумана,  
2022 год

# Пакеты

---

Пакет в Python – это каталог, включающий в себя другие каталоги и модули.

Виды пакетов:

- регулярные (до 3.2)
- пространств имён (3.3+)

Регулярный пакет дополнительно содержит файл `__init__.py`.

Наличие файла `__init__.py` позволяет импортировать модули из каталога.

Переменная `__all__` - список публичных объектов данного модуля, т.е. тех имён, которые можно из него импортировать.

# Переменная `__name__`

---

При запуске текущего модуля содержит значение `__main__`, при импортировании из другого модуля - название текущего модуля.

```
if __name__ == '__main__':  
    print('hello')
```

# Модули distutils, setuptools

---

distutils отвечает за создание *дистрибутивов* - архивов с кодом пакетов. Пакет должен содержать файл setup.py.

setuptools - надстройка над distutils, позволяющая предоставить пакетному менеджеру информацию о пакете и о *зависимостях*.

# Пакетные менеджеры

---

PyPI - Python Package Index, <https://pypi.org/>

pip - менеджер пакетов

pip install some-package-name

pip uninstall some-package-name

pip install -r requirements.txt

# Версии пакетов

---

Уровни версии (по semver):

- мажорная версия - когда сделаны обратно несовместимые изменения API
- минорная версия - когда добавлена новая функциональность, не нарушая обратной совместимости
- патч-версия - когда сделаны обратно совместимые исправления

`docopt == 0.6.1`      `# Version Matching. Must be version 0.6.1`

`keyring >= 4.1.1`      `# Minimum version 4.1.1`

`coverage != 3.5`      `# Version Exclusion. Anything except version 3.5`

`Mopidy-Dirble ~= 1.1`      `# Compatible release. Same as >= 1.1, == 1.*`

# Контроль зависимостей

---

Проблемы зависимостей: пакет **a** зависит от пакета **c** версии  $\leq 1.0.0$ , а пакет **b** зависит от пакета **c** версии  $\geq 1.3.5$

Venv, virtualenv, pipenv, poetry...

# Корутины (сопрограммы)

---

Корутина - генератор, который *принимает* значения.

Оператор `yield` - двухсторонний.



# Асинхронность

---

Процесс (задача) - программа, находящаяся в режиме выполнения.

Поток (thread) - наименьшая единица обработки, исполнение которой может быть назначено ядром операционной системы.

Нить (fiber) - секция исполняемого кода, для которой должна быть вручную установлена очередность обслуживания прикладной программой.

Некоторые языки позволяют выполнять **квазипараллельные** действия в рамках одного потока.

**async** - объявление оператора асинхронным

**await** - ожидание результата асинхронного оператора; возможно переключение на другую сопрограмму

GIL - global interpreter lock

# base64

---

Стандарт кодирования двоичных данных при помощи 64 символов ASCII:  
52 английских буквы, 10 цифр, +, /

$$2^6 = 64$$

**10101010**

$$\text{НОК}(6, 8) = 24$$

То есть каждые 3 байта кодируются 4 символами. В конце закодированной строки может добавляться = или ==

# uuid

---

Universally unique identifier, универсальный уникальный идентификатор - 16-байтный номер.

5 групп цифр в формате 8-4-4-4-12:

123e4567-e89b-12d3-a456-426655440000

Генерируется на основе времени, аппаратных особенностей компьютера и случайных чисел.

UUIDv0, UUIDv1, UUIDv2, ...

GUID - реализация UUID от Microsoft

# XML, JSON

---

XML - eXtensible Markup Language

JSON - JavaScript Object Notation

# XML

---

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- IU7-71 -->
<students>
  <student card_number="18Y998">
    <lastname>Иванов</lastname>
    <firstname>Иван</firstname>
  </student>
  <student card_number="18Y999">
    <lastname>Петров</lastname>
    <firstname>Пётр</firstname>
  </student>
</students>
```

Теги (узлы), атрибуты

# XSD, XPath, XQuery, XSLT...

---

XSD - XML Schema definition (диалект XML со специализированными тегами)

XPath - язык поиска по XML-документу (по принципу построения путей в файловой системе) `/tag1[attribute1='value1']/tag2`

XQuery - язык запросов для модификации XML-документов

XSLT (eXtensible Stylesheet Language Transformations) - язык преобразования XML-документов

В HTML 4.0 - XHTML

# JSON

---

```
{“students”: [  
  {“lastname”: “Иванов”, “firstname”: “Иван”},  
  ...  
]}
```

# JSON Schema, path, query

---

Аналоги схемы XML, языка пути и языка запросов

```
class[-1].marks.informatics
```



# gc

---

Каждый объект хранит в себе количество ссылок (переменных, которые на него ссылаются).

```
sys.getrefcount(foo)
```

Сборщик мусора периодически возвращает блоки памяти, на объекты в которых не осталось ссылок, операционной системе.

Проблема: циклические ссылки. Их невозможно учитывать, требуется дополнительный алгоритм.