

Python



modules and packages, __init__.py, __main__.py, virtualenv,
six, future, past, builtins, pep8, joe

Модуль (Module)

- Модулем является любой валидный Python-файл с расширением ".py"
- В процессе выполнения интерпретатором ".py" модулей, создаются скомпилированные ".pyc" файлы с байт-кодом, можно вызвать принудительно: `python -m compileall .`

Практика

Файл `dis_example.py`

Пакет (Package)

- Пакетом является папка с модулями, у которой есть магический-файл `__init__.py`
- Если файла нет - **пакета не получится!**
- Существуют другие волшебные файлы: `__main__.py`, если он присутствует в пакете, то пакет можно выполнить: `python -m package_name`

Практика

Файлы "exes_module/*.py"

Импортирование пакета или модуля

- Для импорта используются две конструкции:

import module

from package **import** module

- Можно импортировать пакет, модуль, отдельные части модуля: функции, константы, классы

from package.module **import** my_func, VALUE, MyClass

- Импорты бывают абсолютные и относительные, лучше использовать абсолютные.

Практика

Файл "imports_main.py"

Установка НОВЫХ пакетов

- `pip` - система управления пакетами Python, `pip3` - аналог для Python3
- Некоторые установленные пакеты работают только на Python2, некоторые только на Python3, некоторые на двух версиях интерпретатора

```
# активируем виртуальную среду  
source virtual_env/bin/activate
```

```
# устанавливаем пакеты
```

```
    pip install six
```

```
    pip install future
```

```
    pip install pep8
```

```
    pip install joe
```

Существуют библиотеки, которые помогают писать код под две версии Python

- 2to3 и 3to2 (<http://www.diveintopython3.net/porting-code-to-python-3-with-2to3.html>)
- six
- future
- __future__
- другие

Практика

Файл "2to3_examples/*.py"

joe - создаст для
необходимы списки
для .gitignore

<https://github.com/karan/joe>

Как писать код под обе версии?

http://python-future.org/compatible_idioms.html

Декомпозиция сложных задач к простым

Сложная задача: "Написать игру морской бой с
сохранением игры и ИИ"

Какие модели используются в игре?

- Игрок, свойства: очередь, имя
- Поле, свойства: размер, количество кораблей
- Корабль, свойства: размер (длина), место на поле
- Выстрел, свойства: кем, координаты, промах или попадание

Как будем хранить текущие данные?

- Нам нужно хранить: игроков, позиции кораблей, сделанные выстрелы

Какая последовательность действий?

- Создается пустое поле
- Создаются два игрока
- Каждый игрок по очередности правильным образом размещает все свои корабли на поле
- Перед каждым выстрелом отрисовывается новое поле, с известными выстрелами, пустыми пространствами и уничтоженными кораблями
- По очереди каждый игрок делает выстрел, если выстрел приходится в цель, то игрок продолжает стрелять.
- Игра длится, пока все корабли одного из игроков не будут уничтожены