

УРОК 36. МНОГОМОДУЛЬНЫЕ ПРОГРАММЫ, ПАРСИНГ ВЕБ-СТРАНИЦ

МНОГОМОДУЛЬНЫЕ ПРОГРАММЫ	2
ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ	4
ПАРСИНГ ВЕБ-СТРАНИЦ	5
ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ	7
ПРАКТИЧЕСКАЯ РАБОТА	8
ПОЛЕЗНЫЕ МАТЕРИАЛЫ	9



МНОГОМОДУЛЬНЫЕ ПРОГРАММЫ

В Python, при импорте модуля, интерпретатор сначала ищет файл с расширением .py, а затем, если он найден, создает и сохраняет скомпилированный .pyc-файл (байт-код) для более эффективной загрузки модуля в будущем. .pyc-файл содержит предварительно скомпилированный код, который интерпретатор может выполнять непосредственно.

Пакеты в Python являются способом организации и группировки связанных модулей. Пакеты представляют собой директории, которые содержат модули и другие пакеты, а также файлы `__init__.py`, которые указывают интерпретатору, что директория является пакетом.

Менеджер пакетов `pip` является инструментом для установки, управления и обновления сторонних пакетов Python. С помощью `pip` можно легко устанавливать пакеты из репозитория Python Package Index (PyPI) и других источников.

Python

```
# Установка пакета с помощью pip  
pip install package_name
```

Виртуальное окружение (virtual environment) позволяет изолировать проекты Python и управлять их зависимостями. Виртуальное окружение создается для каждого проекта, и пакеты устанавливаются в него, не затрагивая системную установку Python. Это позволяет иметь разные версии пакетов для разных проектов и избегать конфликтов.



Python

Создание виртуального окружения

```
python -m venv myenv
```

Активация виртуального окружения (в UNIX-системах)

```
source myenv/bin/activate
```

Установка пакетов в виртуальное окружение

```
pip install package_name
```

Деактивация виртуального окружения

```
deactivate
```



ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ

Запустите в терминале команду:

```
Python  
pip install bs4
```

Когда установка закончится, поясните, что произошло.



ПАРСИНГ ВЕБ-СТРАНИЦ



Библиотека **Beautiful Soup** является мощным инструментом для парсинга и обработки HTML- и XML-документов. Она позволяет удобно извлекать данные из веб-страниц, навигировать по структуре документа и выполнять различные манипуляции с содержимым.

Python

```
import requests
from bs4 import BeautifulSoup

html = requests.get("https://example.com").text
# Создание объекта BeautifulSoup из HTML-страницы
soup = BeautifulSoup(html, "html.parser")

# Извлечение данных из тегов
title = soup.title.text
links = soup.find_all("a")

# Навигация по структуре документа
parent = soup.find("div").parent
next_sibling = soup.find("div").next_sibling

# Манипуляции с содержимым
new_tag = soup.new_tag("a", href="https://example.com")
soup.body.append(new_tag)
```

HTML (HyperText Markup Language) является стандартным языком разметки для создания веб-страниц. Он определяет структуру и содержимое страницы с помощью тегов, атрибутов и текстового контента.

Парсинг HTML с помощью **Beautiful Soup** включает в себя создание объекта **Beautiful Soup** из HTML-страницы и использование различных методов и атрибутов для



извлечения нужных данных. Можно осуществлять поиск элементов по тегам, классам, идентификаторам, а также выполнять навигацию по структуре документа.

Пример извлечения данных из сырого HTML:

```
Python
from bs4 import BeautifulSoup

# Создание объекта BeautifulSoup из сырого HTML
html = """
<html>
<body>
  <h1>Заголовок</h1>
  <p>Текст параграфа</p>
  <a href="https://example.com">Ссылка</a>
</body>
</html>
"""

soup = BeautifulSoup(html, "html.parser")

# Извлечение заголовка
title = soup.find("h1").text

# Извлечение текста параграфа
paragraph = soup.find("p").text

# Извлечение ссылки
link = soup.find("a")["href"]
```

Таким образом, работа с многомодульными программами и парсингом веб-страниц в Python включает организацию кода в модули и пакеты, использование менеджера пакетов `pip` для установки необходимых библиотек, работу с виртуальными окружениями, а также использование библиотеки `Beautiful Soup` для парсинга и обработки HTML-страниц.



ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ

Что будет выведено в результате выполнения программы:

```
Python
import requests
from bs4 import BeautifulSoup

html = requests.get("https://realpython.com").text
soup = BeautifulSoup(html, "html.parser")

links = soup.find_all("a")
for i in links:
    href = i.attrs.get("href")
    if href[:4] == "http":
        print(href)
```



ПРАКТИЧЕСКАЯ РАБОТА

Написать программу, которая скачивает страницу с сайта на выбор с актуальным курсом валют, выделяет курс доллара к евро и выводит его. Можно пользоваться регулярными выражениями и BeautifulSoup вместе.



ПОЛЕЗНЫЕ МАТЕРИАЛЫ

1. [Облегчаем себе жизнь с помощью BeautifulSoup4](#)
2. [Устанавливаем python-пакеты с помощью pip](#)