

Jinja2

№ урока: 5 **Курс:** Django Essential

Средства обучения: Pycharm Community, GIT, аккаунт в GitHub, Python 3.6 и выше, Django проект, Docker образ.

Обзор, цель и назначение урока

Научиться использовать Jinja2 в своих проектах. Рассмотреть преимущества и недостатки от использования данного инструмента. Узнать, как подключать Jinja2 к проекту, а также изучить отличия от Django шаблонов.

Изучив материал данного занятия, учащийся сможет:

- Подключать Jinja2 к проекту.
- Создать шаблон с помощью Jinja2.
- Создать шаблон наследование шаблонов.
- Подключать backend часть Jinja2.
- Создать собственный backend.

Содержание урока

1. Установка Jinja2
2. Подключение Jinja2 к проекту
3. Рассмотрение программных интерфейсов Jinja2
4. Синтаксис Jinja2. Основные пункты и поиск по документации
 - a. Переменные
 - b. Фильтры
 - c. For
 - d. If
 - e. Macros
 - f. Call
 - g. Set
 - h. Math, Comparisons, Logic
 - i. Python методы
5. Создание базового шаблона и дочернего шаблона
6. Создание своего backend-а под шаблон

Резюме

- Jinja2: как шаблонизатор для Django и как шаблонизатор для документов.
- Для подключения Jinja2 к проекту используем pip и settings.py.
- Программные интерфейсы: загрузчики и окружение.
- Основные методы шаблона с Jinja2: render, generate, stream.
- Синтаксис похож на шаблоны Django но более расширенный.
- Также есть 4 основные типа конструкций.
- Переменные используются с context dictionary. Можно использовать точку (".") для доступа к атрибутам.
- Фильтры реализованы с помощью ключевого знака "|".

- Циклы: итерация по элементам списка и словаря (items()). Также есть возможность использовать if и else.
- В середине цикла есть множество специальных переменных, которые упрощают использование.
- В конструкции "if" можно использовать elif.
- Макросы - функции внутри Jinja2.
- Call используется для вызова функции (макроса).
- С помощью set можно переопределять переменные или задавать им значение.
- Jinja2 поддерживает те же математические знаки что и python (+, -, /, //, %, *, **).
- Логические операции: and, or, not.
- Jinja2 поддерживает множество python методов.
- Для наследования используем block, extends, super, include, import.
- Для создания своего backend шаблонизатор должен наследоваться от django.template.backends.base.BaseEngine и определить метод get_template().

Закрепление материала

- Какие основные цели использования Jinja2?
- В чем отличия Jinja2 от Django шаблонов?
- Какие есть программные интерфейсы в Jinja2?
- Что такое macros в Jinja2?
- Какие инструменты в Jinja2 дают возможность реализовать наследование?

Дополнительное задание

Задание

Сгенерировать html шаблон с помощью Jinja2 с использованием Jinja2 синтаксиса, наследования и своего backend. Рефакторинг предыдущего кода с использованием Jinja2.

Самостоятельная деятельность учащегося

Задание 1

Выучить и понять все преимущества и недостатки Jinja2, как инструмента. Провести аналогию с подобными инструментами.

Задание 2

Создать несколько шаблонов, реализовать между ними наследование и реализовать на дочерних шаблонах несколько типов алгоритмов сортировки ("Пузырьковая сортировка", "Сортировка выбором", "Сортировка вставками").

Задание 3

Сгенерировать с помощью Jinja2 конфигурационный файл на примере nginx.conf.

Рекомендуемые ресурсы

Сайт Jinja2 проекта: <https://jinja.palletsprojects.com/en/2.10.x/>

Пример конфигурации: <https://ruhighload.com/nginx.conf>

Примеры сортировки: <https://webdevblog.ru/algoritmy-sortirovki-v-python/>