

Шаблонизатор Jinja

Jinja — текстовый шаблонизатор, для языка Python, позволяющий генерировать разветку HTML

Django предоставляет собственный шаблонизатор, который очень схож с Jinja 1

{{ value }} - шаблон, для отображения переменных

Внутри данного шаблона, поддерживаются различные методы, по обработки информации

{{ value|method }}

lower — аналогичен методу lower в Python

default:“value“ — значение по умолчанию для переменной, если ее значение пустое

default_id_none:“value“ — то же что и default, но если значение переменной None

length — возвращает длину значения указанной переменной (len)

filesizeformat — приводит значение размера файла к удобочитаемому формату

add:“value“ — производит суммирование значения переменной с указанным значением, так же позволяет суммировать значение двух переменных

addslashes — добавляет косую черту перед кавычками, используется для экранирования, если текст имеет кавычки

capfirst — делает первую букву заглавной

center:“length“ — центрирует значение переменной до указанной длины (center)

cut:“value“ — удаляет из переменной все указанные значения (replace)

date:“format“ — форматирует UNIX дату в соответствии с форматом

dictsort:“key“ — применяется к списку словарей, и сортирует данный список словарей по указанному ключу

dictsortreverse:“key“ — то же что и dictsort, только сортирует в обратном порядке

divisibleby:“value“ — возвращает True, если значение переменной кратно value

escape — экранирует HTML теги внутри значения текстовой переменной

escapejs — то же что и escape, но для экранирования символов JavaScript

first — возвращает первый элемент из списка ([0])

floatformat:2 — округляет дробное число с указанной точностью если тождность не указана, то округляет до 1 символа после запятой

join:“value“ — аналогичен методу join

last — возвращает последний элемент в списке ([-1])

length_is:“value“ — возвращает True, если длина значения переменной равна заданному значению

linebreaks — заменяет значение переноса строки на тег

linenumbers — отображает номера строк текста

ljust:“length“ — выравнивание по левому краю в поле заданной ширины

rjust:“length“ — то же что и ljust, но для выравнивания по правому краю

make_list — преобразовывает значение переменной в список

phone2numeric — преобразовывает номер телефона в числа (если номер телефона содержит буквы)

random — возвращает случайный элемент из списка

slice:“start:stop“ — берет срез из списка

slugify — преобразует текст в slug

time:“format“ — преобразует время в соответствии с форматом

timesince:date — применяется между двумя датами, находит разницу (timedelta)

title — аналогичен методу title в Python

upper — аналогичен методу upper в Python

urlize — преобразует URL и Email в кликабельные ссылки

urlizetruc:length — то же что и urlize, только обрезает отображение ссылки до указанной длины

wordcount — возвращает количество слов

{% %} - шаблон для создания блоков и конструкций

{% block blockname %}{% endblock %} - создает блок, который будет переопределен в дочернем шаблоне, используется для наследования с целью уменьшения кода

{% comment %}{% comment %} - используется для указания комментария в разметке HTML

{% csrf_token %} - генерирует токен для защиты, используется при создании форм

{% extends "path" %} - используется для указания родительского шаблона

{% filter lower|linenumbers %}{% endfilter %} - используется для применения нескольких фильтров к тексту помещенному в данный блок

{% for %}{% endfor %} - используется для создания цикла

{% empty %} - необязательный блок для цикла, для указания текста, если список был пустой

{% if %}{% endif %} - используется для создания условного оператора внутри шаблона

{% include "template.html" %} - используется для включения другого шаблона

{% now "format" %} - отображает текущую дату и время в соответствии с форматом

{% url "url" %} - формирует относительную ссылку

{% load static %} - используется для подключения папки *static*

{% static "path" %} - используется для указания расположения static файла

Управление статическими файлами

Для управления (подключения) статическими файлами, в файле *settings.py* необходимо определить несколько переменных:

STATIC_URL = "static/" - указание к главной папки с статическими файлами

STATICFILES_DIRS = [] - список с указанием дополнительных папок с расположением статических файлов

STATIC_ROOT = "path" — расположение *static* файлов, куда они будут перенесены после выполнения команды **collectstatic** (используется при развертывании проекта)