

Python&Django

Автор курса



Антон Мазун

Full-stack Python developer
Тренер-консультант CBS



Python&Django

Шаблоны и представление

Python&Django

Шаблон Django

Шаблон Django - это просто текстовый файл , или строка Python , которые следуют языку шаблонов Django

Синтаксис языка шаблонов Django использует четыре конструкции.

- **Переменные** : `{{variable}}`
- **Фильтры** : `{{variable|filter }}`
- **Теги**: `{%...%}`
- **Комментарии** : `{# comment #}`

Python&Django

Переменные

Переменные выглядят следующий образом : `{{variable}}`

Он пытается выполнить поисковые запросы в следующем

Порядке:

- Поиск словаря: `foo["bar"]`
- Поиск атрибута или метода : `foo.bar`
- Поиска по цифровому индексу: `foo[bar]`

Если вы используете переменную , которая не существует , система шаблонов будет вставлять значение `string_if_invalid = ""`.

Python&Django

Фильтры

Вы можете изменять переменные для отображения с помощью фильтров. Фильтры выглядят следующим образом :

- `{{variable|filter}}`
- `{{variable|filter : "args"}}`

Пример:

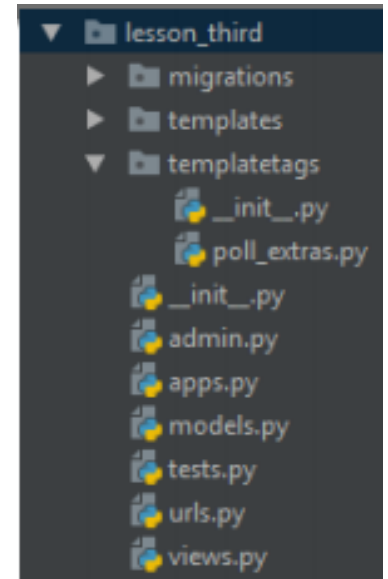
```
Context = {"name" : "JOHN"}
```

```
{{name|lower }}
```

Python&Django

Свои фильтры

- 1) Создать в приложении папку `templatetags` , на одном уровне вложенности с `models.py` , `views.py`
- 2) В файле `poll_extras.py` зарегистрировать свой фильтр.
- 3) Подключить в шаблоне с помощью тега `{% load poll_extras %}`



Python&Django

Теги

Синтаксис : {% tag %} Что-то делать {% endtag %}

- 1) if/else
- 2) for
- 3) comment

Python&Django

IF

В Python и в шаблонной системе Django нижеприведённые объекты вычисляются в False в контексте Boolean:

Пустой список ([]);

Пустой кортеж ();

Пустой словарь ({});

Ноль (0);

Особый объект None;

Объект False;

Можно использовать **and** и **or** вместе. Приоритет **and** больше.

{% if list **and** some_var **or** object%} ---> if (list **and** some_var) **or** object:

Тег if можно вкладывать в if.

Можно использовать операторы:

`==, !=, <, >, >=, <=` и `in`

Python&Django

FOR

{% for x in y %} , x - это имя переменной, в которую помещается значение текущего элемента.

y - это последовательность.

{%endfor%}

for ... empty

{% for x in y %}

.....

{% empty %}

..... - выполняется в случаи , если последовательность пуста

{% endfor%}

Python&Django

Переменные цикла for

forloop.counter	Номер текущей итерации цикла (начиная с 1).
forloop.counter0	Номер текущей итерации цикла (начиная с 0).
forloop.revcounter	Число итераций цикла, оставшихся до окончания перебора (завершая единицей).
forloop.revcounter0	Число итераций цикла, оставшихся до окончания перебора (завершая нулем).
forloop.first	True, если это первая итерация цикла.
forloop.last	True, если это последняя итерация цикла.

Python&Django

Regroup

Группирует список похожих элементов по одинаковому признаку.

Male:

- * George Bush
- * Bill Clinton

Female:

- * Margaret Thatcher
- * Condoleezza Rice

Unknown:

- * Pat Smith

Если нужные данные представлены в виде списка словарей, то можно отсортировать данные прямо в шаблоне, применив фильтр **dictsort**.

Наследование шаблонов

Python&Django

Компоненты шаблонной системы

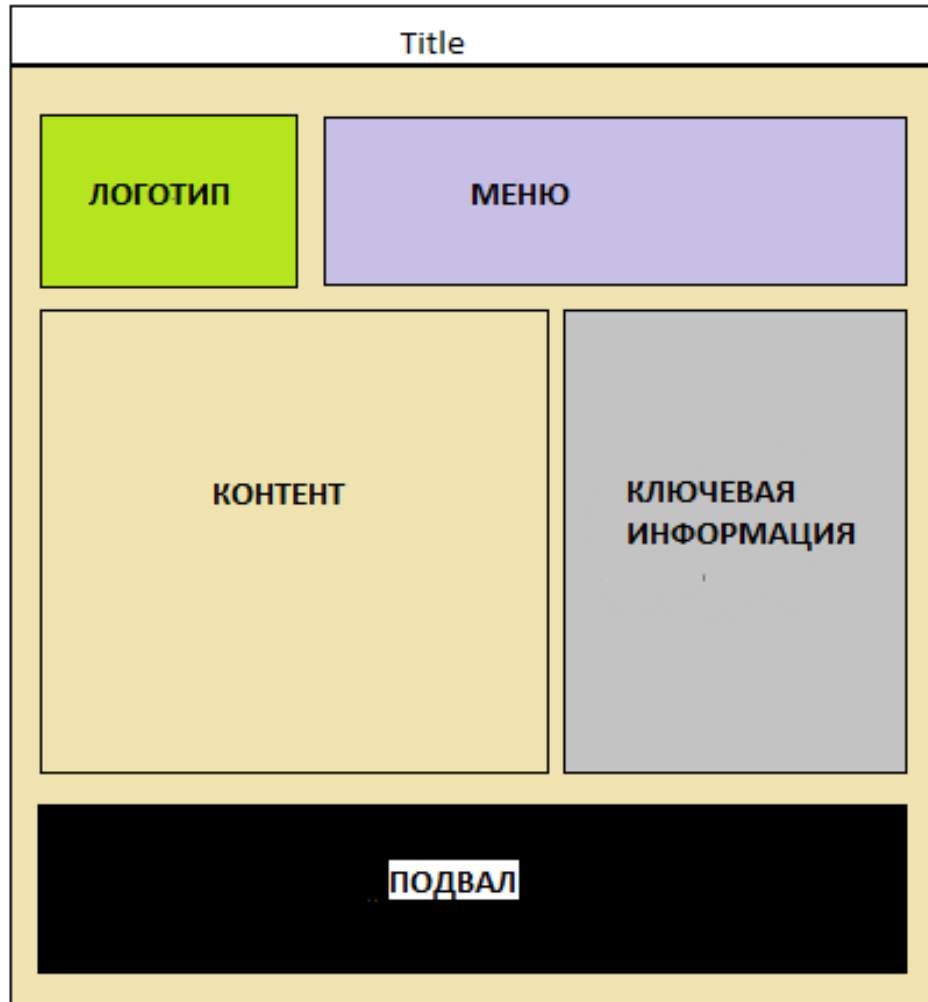
`{% include "some.html" %}` - подставляет запрошенный шаблон (отрендеренный в текущем контексте).

`{% block name %}...{% endblock %}` – блочный элемент

`{% extend %}` - указываем, какой шаблон мы будем уточнять.

Python&Django

Применение шаблонной системы



```
{% block title%} Title {% endblock%}
<body>
    <header>logo, menu </header>
{% block content%}    {% include ".html"%}
    Статьи и тд...
{% endblock%}

    <footer> FOOTER</footer>
</body>
```

Спасибо за внимание!

Информационный видеосервис для разработчиков программного обеспечения

