

Практическое задание по шаблонизации

Часть 1

В этом задании вы научитесь формировать данные для шаблона (контекст), передавать эти параметры в шаблон и изучите базовые основы языка шаблонизации **Django**.

Для выполнения задания необходимо использовать шаблон проекта

https://github.com/alexopryshko/coursera_assignment_tmp

В **views.py** нужно создать функцию **echo(request)**, которая принимает **request** и возвращает **HttpResponse** всегда со статусом **200**, а также она должна возвращать эхо, переданных в запросе, параметров и значение заголовка **X-Print-Statement**. **View** должна быть доступна по пути **/template/echo/**

View которые необходимо отредактировать находятся в **template/views.py**, **template - template/templates/echo.html**.

Решение должно быть **zip** архивом. В архиве должно содержаться два файла, которые вы редактировали ранее (обратите внимание, название файлов должно быть именно таким): **views.py**, **echo.html**

Примеры запросов и ответов:

Метод	Параметры	Заголовки	Ответ
get	GET параметры: a=1	-	get a: 1 statement is empty
get	GET параметры: c=2	-	get c: 2 statement is empty
post	POST параметры: b=1	-	post b: 1 statement is empty
post	POST параметры: d=3	-	post d: 3 statement is empty
get,post	-	X-Print-Statement = test	statement is test
get,post	-	-	statement is empty

Замечание: заголовки находятся в **META** параметрах запроса. <https://docs.djangoproject.com/en/2.0/ref/request-response/#django.http.HttpRequest.META>

Часть 2

В этом задании вы научитесь создавать темплейт фильтры. Фильтры часто используются на практике, например, когда необходимо привести данные к конкретному виду (<https://docs.djangoproject.com/en/1.11/ref/templates/builtins/>), в частности:

- изменить формат даты
- Привести строку к нижнему регистру

Для выполнения задания необходимо использовать шаблон проекта

https://github.com/alexopryshko/coursera_assignment_tmp

Templatetags которые необходимо отредактировать находятся в **template/templatetags/extras.py**. Решение должно быть файлом **extras.py**.

2.1. Фильтр **inc**. Необходимо в файле **extras.py** создать фильтр **“inc”** который принимает 2 аргумента: 1-й - число которое нужно увеличить, 2-й - на сколько нужно увеличить первое число. Пример использования фильтра **“inc”** представлен в файле **template/templates/filters.html**

2.2. Тег **division**. Необходимо в файле **extras.py** создать тег **“division”** (то есть тег для деления), который принимает 3 параметра: 1-ый - делимое, 2-ой - делитель, 3-ий — флаг определяющий тип возвращаемого значения для результата деления (именованный аргумент **to_int**). Если переданное значение **to_int** равно **False**, необходимо выполнить вещественное деление. Если передано **True** результат вещественного деления необходимо привести к целому. Значение

to_int по-умолчанию — False.

Обратите внимание, что делимое и делитель целые числа, но передаются в тег в формате string.

Пример использования тега “division” представлен в файле `template/templates/filters.html`

```
1 {% division a b to_int=True %}  
2 {% division a b %}
```

Часть 3

В этом задании вы научитесь работать с наследовании шаблонов. Наследование повсеместно используется. Например, header и footer у сайта одинаковый, не стоит дублировать код header и footer на каждой странице, имеет смысл сделать базовый шаблон и наследоваться от него.

Для выполнения задания необходимо использовать шаблон проекта

https://github.com/alexopryshko/coursera_assignment_tmp

Шаблон, который необходимо отредактировать **template/templates/extend.html**. Решение должно быть файлом **extend.html**.

Нужно написать шаблон которые наследуется от **template/templates/base.html** и переопределяет блоки `block_a`, `block_b`. В блоках `block_a`, `block_b` должно остаться, то что было у родительского шаблона и дополнительно вывести параметр “a” и параметр “b”