

ДЗ#4: JS

Основы веб-разработки (первый семестр)

Здесь о том, что касается javascript в рамках ДЗ#4.

Итак. Задача - есть три фичи (с)

- формочка в диалоговом окне (делали на лекции)
- автообновление списка объектов (делали на лекции)
- кнопка лайка одного из объектов в списке, работающая через ajax

Берем любые две и реализуем в проекте.

Если вам ни одна из этих не нравится, и вы хотите сделать что-либо другое - пожалуйста, я совершенно не против. Больше разнообразия - это безусловно хорошо! :)

Как я буду оценивать эту задачу:

Я буду оценивать её без жести, по двум критериям - корректность javascript-кода и собственно работоспособность самой фичи :)

jQuery, jQueryUI

Стандартная библиотека javascript довольно скудна, и делать на голом js что-либо комплексное - ад, боль и унижение. Поэтому есть немаленькое количество js-фреймворков, которые здорово облегчают жизнь. Стандарт де-факто сейчас пожалуй jquery и jqueryui.

jQuery

Это библиотека, реализующая огромное количество всевозможного функционала.

Безусловно, в живой разработке вы столкнетесь именно с jquery в качестве основного инструмента для оживления проекта.

Домашний сайт библиотеки - <https://jquery.com/>

Кроме того, по интернету раскидана гора справочников, статей и tutorиалов.

<http://jquery.page2page.ru/> очень рекомендую. Так же, как и с bootstrap-ом - подключить можно как с cdn-а, так и просто скачав себе библиотеку.

Для jQuery существует огромное количество плагинов, реализующих тот или иной функционал на базе этой библиотеки. Например, очень рекомендую

<https://harvesthq.github.io/chosen/>

Примеров - огромное количество, как и в документации так и просто раскидано по интернету.

jQueryUI

Надстройка над jQuery, включающая некоторое количество дополнительных виджетов.

Основной сайт - <https://jqueryui.com/>, там же инструкция по установке. Кроме того,

Прямой эфир

Марина Данышина 19 минут назад

Расписание занятий → **Расписание пересдач** 0

Екатерина Рогошкова 9 часов назад

Получение значков достижений на портале (ачивки) 2

Ольга Августан Вчера в 18:41

Опросы → **Получение значков достижений на портале (ачивки)** 2

Ольга Августан Вчера в 13:19

Общие вопросы → **Завершение семестра: последние итоговые встречи** 0

Ольга Августан Вчера в 13:12

Завершение семестра: итоговые встречи по дисциплинам 10

Марина Данышина 25 Мая 2016, 14:01

Разработка на Java (первый семестр) → **Забыли зонтик** 0

Ksenia Sternina 25 Мая 2016, 13:38

Домашнее задание - Юзабилити-тестирование 4

Летяго Владимир 25 Мая 2016, 01:40

Разработка приложений на Android - 1 (второй семестр) → **Контрольный рубеж** 0

Летяго Владимир 24 Мая 2016, 22:45

Домашнее задание №3 10

Вячеслав Ишутин 24 Мая 2016, 22:31

Разработка на C++ (открытый курс) → **Итоговая встреча по дисциплине** 0

Вячеслав Ишутин 24 Мая 2016, 22:22

Разработка на C++ (открытый курс) → **Итоговое занятие. Презентация курсовых проектов.** 0

Александр Агафонов 24 Мая 2016, 19:17

Мастер-класс Ильи Стыценко онлайн «Использование OAuth 2 в приложениях на Django» 1

Марина Данышина 23 Мая 2016, 20:59

Мероприятия → **Мастер-класс Ильи Стыценко онлайн «Использование OAuth 2 в приложениях на Django»** 1

Ольга Августан 23 Мая 2016, 17:56

Разработка приложений на iOS - 1 (второй

на <http://jquery.page2page.ru/> тема jqueryui тоже подробно освещена.

Примеры виджетов - диалоговое окно, datepicker (календарик для выбора дат), autocomplete (замена стандартному select-у, спасающая в случае большого количества вариантов для выбора).

Вопросы вы задаете обычно в последний момент, так что попробую новый формат - придумаю вопросы за вас, и отвечу на них :)

А что, всплывающую формочку делать прямо так, как в скринкасте?

Да не, та жесть была в обучающих целях. Есть куча плагинов, как в bootstrap так и в jqueryui. Берите то, что вам ближе.

Вы заставляете меня выносить весь js в отдельный файл, а как же мне сделать что-то с конкретным элементом, или передать в скрипт какой-либо параметр из шаблона???

Для этого используйте data-атрибуты, сейчас покажу как.

У любого html-элемента есть возможность определить некоторое количество атрибутов, начинающихся с data-. Это специальное подмножество атрибутов, выделенное специально для того, чтобы их содержимое читалось js-скриптами. То есть, с помощью этих атрибутов можно сконфигурировать поведение того или иного элемента.

На примере:

Допустим, у нас на странице есть поле ввода для даты. По умолчанию оно не очень удобно, и мы хотим сделать из него няшный календарик средствами jqueryui (вот он <https://jqueryui.com/datepicker/>).

Допустим, наш собственный js-скрипт для проекта (он у нас пока один) лежит в base.js.

Что мы делаем - прежде всего нам нужно заставить яваскрипт найти этот элемент на странице. Следите за руками -

(в шаблоне)

```
<input type="text" name="mydate" data-widget-type="datepicker">
```

(в js-файле)

```
$('#[data-widget-type=datepicker]').datepicker();
```

Что мы только что сделали - нашли все элементы с атрибутом data-widget-type

семестр) → Завершение семестра 0

Ольга Августан 23 Мая 2016, 17:16

Итоговая встреча по дисциплине 2

Ольга Августан 23 Мая 2016, 17:07

Основы веб-разработки (первый семестр) → Итоговая встреча по дисциплине 0

Ольга Августан 23 Мая 2016, 17:06

Основы мобильной разработки (первый семестр) → Итоговая встреча по дисциплине 2

Ольга Августан 23 Мая 2016, 17:05

Проектирование интерфейсов мобильных приложений (второй семестр) → Итоговая встреча по дисциплине 0

Ольга Августан 23 Мая 2016, 17:02

Проектирование СУБД (второй семестр) → Итоговая встреча по дисциплине 0

Ольга Августан 23 Мая 2016, 16:59

Общие вопросы → Завершение семестра: итоговые встречи по дисциплинам 10

Весь эфир

Блоги

Основы веб-разработки (первый семестр)	26,04
Разработка на C++ (открытый курс)	16,97
Основы мобильной разработки (первый семестр)	14,70
Разработка на Java (первый семестр)	13,58
Общие вопросы	5,71
Разработка приложений на Android - 1 (второй семестр)	5,66
Стажировка	4,52
Проектирование интерфейсов мобильных приложений (второй семестр)	3,42
Разработка приложений на iOS - 1 (второй семестр)	2,29
Проектирование СУБД (второй семестр)	2,26

равным `datepicker` и вызвали для них метод `datepicker` (данный метод находится в составе `jqueryui` и расширяет возможности стандартного `jquery`).

Кроме того, можно проитерироваться по всем найденным элементам и для каждого из них сделать что-либо, это делается так:

```
 $('[data-widget=datepicker]').each(function() {  
    console.log(this)  
});
```

Здесь мы у найденного множества элементов вызываем метод `each`, он делает следующее - берет функцию, переданную ему в качестве параметра, и вызывает её для каждого отдельного элемента. Причем внутри этой функции сам очередной элемент доступен нам в переменной `this`.

Кстати, обратите внимание на вызов `console.log` - это способ посмотреть что-либо прямо в консоли браузера (в нашем случае переменная `this` появится во вкладке `console`, а про консоль браузера есть в скринкасте).

Итак, перепишем немножк более комплексно:

```
 $('[data-widget=datepicker]').each(function() {  
    $(this).datepicker();  
});
```

Так мы пробежались по всем элементам с атрибутом `data-widget=datepicker` и для каждого вызвали метод `datepicker`. Конструкция `$(this)` - это получение обертки `jquery` над самим `html`-элементом.

Еще пример:

Допустим, мы прикручиваем то самое автообновление комментариев. То есть - у нас есть `div`, в который нужно время от времени загружать что-либо. В моем примере вызов скрипта происходит прямо в коде `html`-страницы. Вам же нужно вынести его в ваш `js`-файл. У вас возникнет проблема - ссылку, с которой качаются комментарии, вы можете получить только в `django`-шаблоне. А статические файлы через `django`-шаблонизатор не проходят, они передаются браузеру "как есть". А значит ни о каком теге `{% url %}` там не может быть и речи. Решение получается примерно вот такое:

(в шаблоне)

```
<div data-update-from="{% url... %}"></div>
```

(в `js`-файле)

```
 $('[data-update-from]').each(function() {  
    var url = $(this).data('update-from');  
    ура, теперь мы в переменной url знаем нужную ссылку, можем включать код из г  
});
```

Что мы сделали - выбрали все элементы, где вообще присутствует атрибут `data-update-from`. Пробежались по ним всем - и для каждого получили значение атрибута с помощью встроенного метода `.data(имя-атрибута)`. Таким образом мы

вытащили для каждого элемента нужный параметр, который мы ставили ему еще на этапе html-шаблона django. И можем делать то, что нам надо.

Так вы же запретили писать элементы ввода форм чистым html, только через forms. Как же мне добавить атрибут к полю?

Об этом читайте в соседнем посте про верстку.

Я хочу прикрутить лайки, а как же мне сделать что-то по клику на элемент?

Допустим, у нас есть div, в котором валяется текущее количество лайков, и мы хотим при клике на него передавать на сервер новый лайк. Как совершить саму передачу, понятно из предыдущего пункта - мы просто грузим новый html в этот div, а сервер при выдаче этого html добавляет в нужную модельку еще один лайк. Как же нам сделать так, чтобы при клике на этот div начинала работать какая-либо функция?

На примере

(в шаблоне)

```
<div data-click-action="{% url ... %}">{{ post.likes_count }}</div>
```

(в js)

```
$(document).on('click', '[data-click-action]', function() {  
    console.log(this);  
    проверяем, что работает  
    return false;  
});
```

Что мы сделали -

Во-первых, в data-click-action положили тот самый url, который добавит единичку, сохранит объект в базу и вернет нам новое количество лайков для объекта (реализация его - на вашей совести :)

Во-вторых, есть метод on - принимает три параметра -

название события (или event) - в нашем случае click

селектор элементов - для каких элементов метод должен сработать, а для каких - нет; в нашем случае - для всех элементов с атрибутом data-click-action

callback - функция, которая будет вызвана для отдельного элемента в случае возникновения события. внутри - сам элемент в переменной this.

Сам метод on вызывается для того элемента, внутри которого мы вообще хотим переопределить событие.

В нашем случае мы объявляем, что хотим слушать события типа click внутри элемента document (это вся страница в целом). И если событие произошло у элемента, который попадает под фильтр "[data-click-action]" - вызывать данную функцию. Внутри функции мы делаем просто console.log(this) - чтобы проверить, работает ли оно. То есть, пока просто печатаем элемент, на который кликнули. Кто напишет в комментах, зачем return false в конце и где он нам может понадобиться - тому плюс балл :)

Короче говоря, вот так мы на определенные элементы вешаем собственную реакцию на клик пользователем. Внутри функции-callback-а мы можем делать все, что хотим, в том числе загружать для этого элемента новое содержимое. Как именно это делается - было на лекции, а значит есть в скринкасте ;)

А мне прям вот брать base.js и писать туда все, что я хочу сделать?

Не совсем. Все действия с javascript нужно производить только после изначальной загрузки документа. В jquery для этого есть хелпер - функция вот такого вида:

```
$(function () {  
  
    console.log('lalala');  
    $('[data-widget-type=datepicker]').datepicker();  
    все остальное...  
  
});
```

В этом примере мы заворачиваем всё, что хотим сделать, в еще одну функцию. И такой вызов гарантирует нам, что наши действия будут выполнены только после того, как браузер отрисует все элементы на странице. Это важно.

И всё? А есть еще рекомендации?

Есть - включайте ваши собственные css и js-файлы после загрузки bootstrap или jquery. Больше рекомендаций пока не придумалось.

У меня есть вопрос!

Задай его! :)