

Лабораторная работа №6

Тема: " функциональные возможности библиотек JavaScript "

Преподаватель: Атрощенко Н.А.

Цель: освоить основные навыки использования возможностей библиотек JavaScript при проектировании динамических интерактивных веб-страниц.

Основные требования:

- Проект должен включать в себя обязательные папки: соответственно для css, html, resource файлов.
- Ключевой файл должен иметь название index.html.
- На веб-страницах должны быть определены блоки header, footer и основное содержание таблицы.
- Использовать возможности библиотеки JQuery (Prototype, Dojo и др. можно использовать только в качестве дополнения по желанию), запрещается использование других возможностей (DHTML, PHP, gif, flash и т.д.).
- Для второй части задания используйте возможности библиотеки JQuery+Ajax.
- В проекте должен быть единый стиль оформления и определённое тематическое оформление, поэтому картинки и веб-элементы на странице подбирайте соответственно с тематикой и стилем оформления.
- Обязательным должно быть стилевое решение **для всех элементов дизайна веб-страниц**.

Примечание:

приветствуется творческий подход к работе и оформление веб-страниц самостоятельно созданными элементами с индивидуальным дизайном: кнопками, маркерами списков, шрифтами, текстовыми полями и т.д.

Методические указания:

Выполните 2 задания:

Задание А нацелено на развитие навыков создания динамических элементов веб-страниц с помощью библиотеки JQuery и др.

В задании В необходимо реализовать подгрузку содержимого на веб-страницу из указанных файлов (они должны лежать в отдельной папке) и привязку заданных действий к определённым событиям.

Задание А

1. Используя возможности Jquery, создайте следующий сценарий: Создайте таблицу с записями и картинками, расположенными в разных ячейках, а также 2 кнопки. По нажатию на первую кнопку весь текст в чётных строках таблицы должен поменять цвет и размер, а картинки должны быть обведены в рамочку заданной толщины и цвета, а по нажатию на вторую кнопку в нечётных строках таблицы все картинки должны стать постепенно в 2 раза меньше по размеру, а текст заменится на одинаковое слово.
2. Используя возможности Jquery, создайте следующий сценарий: на странице задан текст с 3 параграфами и 3 картинками, а также 2 кнопки. По нажатию первой кнопки в первом параграфе текста меняется его шрифт на курсив, а цвет становится красным. При этом все элементы второго параграфа обводятся в рамочку (в том числе и картинка), а 3 параграф добавится в конец ещё раз (клонируется- clone). По нажатию второй кнопки все 3 картинки станут постепенно невидимыми.
3. Используя возможности Jquery, создайте следующий сценарий: Создайте таблицу с записями и картинками, расположенными в разных ячейках, а также 2 кнопки. По нажатию на первую кнопку найти (используя фильтры) все ячейки таблицы, в которых есть текст,

- сделать шрифт текста курсивом, а также подчеркнуть все элементы `div-a`, содержащие подстроку “LONDON”, а по нажатию на другую кнопку все картинки заданной ширины постепенно заменяются на заданную другую.
4. Используя возможности Jquery, создайте следующий сценарий: на странице задан текст с 3 параграфами и 3 картинками, а также 2 кнопки. По нажатию первой кнопки первый параграф исчезает (постепенно), а 2 остальных параграфа поднимаются вверх и занимают его место. По нажатию второй кнопки все 3 картинки обводятся рамочкой определённого цвета и толщины.
 5. Используя возможности Jquery, создайте следующий сценарий: Создайте таблицу с текстами и заголовками, а также картинками, расположенными в разных ячейках, а также 2 кнопки. По нажатию на первую кнопку весь текст в чётных ячейках должен поменять шрифт на курсив и заголовки типа `h3` должны поменять свой цвет, а картинки заданной высоты (используйте фильтр) во всех ячейках должны постепенно уменьшиться по высоте вдвое.
 6. Используя возможности Jquery, создайте следующий сценарий: на странице задан текст с 3 параграфами и 3 картинками, а также 2 кнопки. По нажатию первой кнопки в первом параграфе текста добавляется смайл после каждого слова ‘new’, а цвет текста второго параграфа становится красным. При этом все элементы третьего параграфа обводятся в рамочку (в том числе и картинка) По нажатию второй кнопки вторая и первая картинки станут невидимыми (постепенно).
 7. Используя возможности Jquery, создайте следующий сценарий: на форме разместить 5 текстовых полей, ненумерованный список и 2 кнопки. По нажатию первой кнопки во всех нечётных текстовых полях (1,3,5) текст должен измениться по цвету и виду шрифта, а в чётных (2,4) текст меняется местами. По нажатию второй кнопки меняется цвет надписи на первой кнопке, а в ненумерованном списке последовательно становятся невидимыми (постепенно) все элементы списка.
 8. Используя возможности Jquery, создайте следующий сценарий: на странице задан текст с 3 параграфами и 3 картинками, а также 2 кнопки. По нажатию первой кнопки в первом и втором параграфе текста меняется его шрифт на курсив, а цвет становится синим. При этом все элементы второго параграфа обводятся в рамочку (в том числе и картинка), а 3 параграф исчезает. По нажатию второй кнопки 3 картинка медленно увеличивается в размере до выбранной величины (постепенно).
 9. Используя возможности Jquery, создайте следующий сценарий: на форме разместить 5 текстовых полей, нумерованный список (5 элементов) и 2 кнопки. По нажатию первой кнопки в текстовых полях (1,3) текст должен измениться и стать таким же, как в 5 поле, а в полях 2,4 измениться по цвету и виду шрифта. По нажатию второй кнопки в ненумерованном списке последовательно становятся невидимыми (постепенно) все нечётные элементы списка.
 10. Используя возможности Jquery, создайте следующий сценарий: на странице задан текст с 3 параграфами и 3 картинками, а также 2 кнопки. По нажатию первой кнопки в первом параграфе текста меняется его шрифт на курсив, а 2 первые картинки исчезают. При этом все элементы второго параграфа обводятся в рамочку заданного цвета и толщины. По нажатию второй кнопки внизу под текстом каждый раз добавляется первая картинка (справа от предыдущей).
 11. Используя возможности Jquery, создайте следующий сценарий: на форме разместить 5 текстовых полей, нумерованный список и 2 кнопки. По нажатию первой кнопки во всех нечётных текстовых полях (1,3,5) текст должен измениться по цвету и виду шрифта, а в чётных (2,4) текст стать невидимым (постепенно). По нажатию второй кнопки меняется цвет текст надписи на первой кнопке, а нумерованный список возрастает в 2 раза (клонируется- `clone`).
 12. Используя возможности Jquery, создайте следующий сценарий: на странице задан текст с 3 параграфами и 3 картинками, а также 2 кнопки. По нажатию первой кнопки в первом параграфе текст увеличивается в 2 раза, а цвет становится синим. При этом все элементы

второго параграфа исчезают, а в конец 3 параграфа добавится картинка из 1 параграфа. По нажатию второй кнопки все 3 картинки станут меньше в 2 раза (постепенно).

13. Используя возможности Jquery, создайте следующий сценарий: на форме разместить 5 текстовых полей, ненумерованный список и 2 кнопки. По нажатию первой кнопки во всех нечётных текстовых полях (1,3,5) в тексте после каждого контента добавиться заданный текст (after(content)), а чётные поля исчезают (постепенно). По нажатию второй кнопки в ненумерованном списке становятся невидимыми первый и последний элементы списка.
14. Используя возможности Jquery, создайте следующий сценарий: на странице задан текст с 3 параграфами и 3 картинками, а также 2 кнопки. По нажатию первой кнопки текст первого параграфа заменяется на картинку 3 параграфа. При этом все элементы второго параграфа обводятся в рамочку (в том числе и картинка). По нажатию второй кнопки все 3 картинки станут постепенно невидимыми и исчезнет 1 кнопка.

Задание В

1. Создайте сценарий с AJAX запросами,, которые должны после нажатия на соответствующие кнопки обращаться к файлам autumn.html, winter. html, summer. html, и spring.html, отображающим соответствующую тематику, загружать их и затем отображать, какой был сделан запрос в текстовое поле. Для выбранной тематики в качестве дополнения должны подгружаться по нажатию на кнопку, используя метод getJSON, данные из соответствующего файла (например, winter. json с 3 характеристиками, одна из которых - массив) и выводиться с подгружением (load) в определённый блок содержимого отдельного текстового файла (например, с реквизитами компании) к уже размещённой информации. Если информация считалась некорректно, используя метод bind, организуйте мигающее предупреждение об этом.

2. Создайте форму с 2 текстовыми полями, выпадающим списком (цвета текста), 3 радиокнопками (размер шрифта). Разместите 3 кнопки button. При нажатии на первую кнопку должны подгружаться по нажатию на кнопку данные из XML—файла с использованием метода jQuery.post и форма должна заполняться данными значениями. По нажатию второй кнопки выполняется локальный скрипт с применением метода jQuery.getScript (с подгружением (load) в определённый блок содержимого отдельного текстового файла (например, с реквизитами компании) к уже размещённой информации), 3-я кнопка должна очистить содержимое формы. Если информация считалась некорректно, используя метод bind, организуйте медленно исчезающее предупреждение об этом.

3. Создайте сценарий с AJAX запросами,, которые должны после нажатия на соответствующие кнопки обращаться к файлам group1.txt, group2.txt, group3.txt, и group4.txt, отображающим соответствующий перечень фамилий, загружать их и затем отображать, какой был сделан выбор, в абзац с id=par. Для выбранной группы в качестве дополнения должны подгружаться по нажатию на кнопку, используя метод getJSON, данные из соответствующего файла (например, group1. json) с 3 характеристиками, одна из которых - массив и выводиться с подгружением (load) в определённый блок содержимого отдельного html файла (например, с реквизитами вуза) к уже размещённой информации. Если информация считалась некорректно, используя метод bind, организуйте мигающее предупреждение об этом.

4. Создайте форму с 3 текстовыми полями (в 1 поле вводится год рождения) и полем кнопок checkbox (цвета текста). Разместите 4 кнопки button. При нажатии на первую кнопку должны подгружаться по нажатию на кнопку данные из XML—файла с использованием метода jQuery.post и форма должна заполняться данными значениями. По нажатию второй кнопки выполняется локальный скрипт с применением метода jQuery.getScript (появляется модальное окно с с подгружением (load) в определённый блок содержимого отдельного текстового файла (например, с реквизитами компании) с отображением сегодняшней даты). По нажатию 3 кнопки все

заполненные данные отображаются в отдельном модальном окне. Если есть незаполненные поля, используя метод bind, организуйте медленно исчезающее предупреждение об этом.

5. Создайте сценарий с AJAX запросами,, которые должны после нажатия на соответствующие кнопки обращаться к файлам city1.html, city2.html, city3.html, и city4.html, отображающим соответствующую тематику, загружать их и затем отображать, какой был сделан запрос в модальном окне. Для выбранной тематики в качестве дополнения должны подгружаться по нажатию на кнопку, используя метод getJSON, данные из соответствующего файла (например, city1.json с 3 характеристиками, одна из которых - массив) и выводиться с подгружением (load) в определённый блок содержимого отдельного текстового файла (например, с реквизитами компании) к уже размещённой информации. Если информация считалась некорректно, используя метод bind, организуйте мигающее предупреждение об этом.

6. Создайте форму с 3 текстовыми полями (в 1 поле вводится год рождения) и полем кнопок radiobutton(цвета текста). Разместите 3 кнопки button. При нажатии на первую кнопку должны подгружаться по нажатию на кнопку данные из XML—файла с использованием метода jQuery.post и форма должна заполняться данными значениями. По нажатию второй кнопки выполняется локальный скрипт с применением метода jQuery.getScript (появляется модальное окно с подгружением отдельного html-файла). По нажатию 3 кнопки отображается возраст отдельном текстовом поле. Если есть незаполненные поля, используя метод bind, организуйте постепенно исчезающее предупреждение об этом.

7. Создайте сценарий с AJAX запросами,, которые должны после нажатия на соответствующие кнопки обращаться к файлам Android OS.html, iPhone OS. html, Bada OS. html, и Symbian OS.html, отображающим соответствующую тематику, загружать их и затем отображать, какой был сделан запрос в текстовой строке. Для выбранной тематики в качестве дополнения должны подгружаться по нажатию на кнопку, используя метод getJSON, данные из соответствующего файла (например, iPhone OS. json с 3 характеристиками, одна из которых - массив) и выводиться с подгружением (load) в определённый блок содержимого отдельного текстового файла (например, с реквизитами компании) к уже размещённой информации. Если информация считалась некорректно, используя метод bind, организуйте мигающее предупреждение об этом.

8. Создайте форму с 2 текстовыми полями (в 1 поле вводится сумма в валюте, во второе – обменный курс), 2 радиокнопками(размер шрифта). Разместите 3 кнопки button. При нажатии на первую кнопку должны подгружаться по нажатию на кнопку данные из XML—файла с использованием метода jQuery.post, и форма должна заполняться данными значениями. По нажатию второй кнопки выполняется локальный скрипт с применением метода jQuery.getScript (появляется модальное окно с подгружением отдельного txt-файла). По нажатию 3 кнопки появляется отдельное текстовое поле с введённой суммой, пересчитанной по курсу. Если есть незаполненные поля, используя метод bind, организуйте медленно исчезающее предупреждение об этом.

9. Создайте сценарий с AJAX запросами,, которые должны после нажатия на соответствующие кнопки обращаться к файлам Belarus.txt, UK.txt, USA.txt, и Russia.txt, отображающим соответствующее описание страны, загружать их и затем отображать, какой был сделан запрос, в абзац с id=par. Для выбранной тематики в качестве дополнения должны подгружаться по нажатию на кнопку, используя метод getJSON, данные из соответствующего файла (например, USA.json) с 3 характеристиками, одна из которых - массив и выводиться с подгружением (load) в определённый блок содержимого отдельного текстового файла (например, с реквизитами компании) к уже размещённой информации. Если информация считалась некорректно, используя метод bind, организуйте мигающее предупреждение об этом.

10. Создайте форму с 3 текстовыми полями (в поля вводятся названия 3 стран) и выпадающим списком (вид отдыха). Разместите 3 кнопки button. При нажатии на первую кнопку должны подгружаться по нажатию на кнопку данные из 3 XML—файлов с использованием метода

`jQuery.post` и в модальном окне должна появится форма, заполненная данными значениями по данным странам. По нажатию второй кнопки выполняется локальный скрипт с применением метода `jQuery.getScript` (появляется модальное окно с отображением последовательно меняющихся 3 картинок). Третья кнопка должна очистить содержимое формы. Если есть незаполненные поля, используя метод `bind`, организуйте мигающее предупреждение об этом.

11. Создайте сценарий с AJAX запросами,, которые должны после нажатия на соответствующие кнопки обращаться к файлам `Minsk.txt`, `Moskow.txt`, `London`, и `Paris.txt`, отображающим соответствующее описание страны, загружать их и затем отображать, какой был сделан запрос, в абзац с `id=par`. Для выбранной тематики в качестве дополнения должны подгружаться по нажатию на кнопку, используя метод `getJSON`, данные из соответствующего файла (например, `Paris.json`) с 3 характеристиками, одна из которых - массив и выводиться с подгружением (`load`) в определённый блок содержимого отдельного html файла (например, с реквизитами компании) к уже размещённой информации. Если информация считалась некорректно, используя метод `bind`, организуйте мигающее предупреждение об этом.

12. Создайте форму с 4 текстовыми полями (имя, адрес, возраст, должность), 3 радиокнопками (размер шрифта). Разместите 4 кнопки `button`. При нажатии на первую кнопку должны подгружаться по нажатию на кнопку данные из XML—файла с использованием метода `jQuery.post` и форма должна заполняться данными значениями. По нажатию второй кнопки выполняется локальный скрипт с применением метода `jQuery.getScript` (появляется файл `swf`, `avi`, или `gif`). 3-я кнопка должна очистить содержимое формы. По нажатию 4 кнопки вся заполненная тем или иным способом информация отображается в отдельном модальном окне. Если есть незаполненные поля, используя метод `bind`, организуйте мигающее предупреждение об этом.

13. Создайте сценарий с AJAX запросами,, которые должны после нажатия на соответствующие кнопки обращаться к файлам `Apple.html`, `Samsung.html`, `Nokia.html`, и `Sony.html`, отображающим соответствующую тематику, загружать их и затем отображать, затем отображать, какой был сделан запрос, в абзац с `id=par`. Для выбранной тематики в качестве дополнения должны подгружаться по нажатию на кнопку, используя метод `getJSON`, данные из соответствующего файла (например, `iPhone OS.json`) с 3 характеристиками, одна из которых - массив и выводиться с подгружением (`load`) в определённый блок содержимого отдельного текстового файла (например, с реквизитами компании) к уже размещённой информации. Если информация считалась некорректно, используя метод `bind`, организуйте мигающее предупреждение об этом.

14. Создайте форму с 4 текстовыми полями (имя, пол, группа, факультет). Разместите 4 кнопки `button`. При нажатии на первую кнопку должны подгружаться по нажатию на кнопку данные из XML—файла с использованием метода `jQuery.post`, и форма должна заполняться данными значениями. По нажатию второй кнопки выполняется локальный скрипт с применением метода `jQuery.getScript` (появляется модальное окно с подгружением отдельного html-файла). Третья кнопка должна очистить содержимое формы. По нажатию четвёртой кнопки все заполненные данные отображаются в отдельном модальном окне. Если есть незаполненные поля, используя метод `bind`, организуйте медленно исчезающее предупреждение об этом.