# Лабораторная работа № 1

## ОСНОВНЫЕ ПОНЯТИЯ И ВОЗМОЖНОСТИ БИБЛИОТЕКИ jQuery.

## СЕЛЕКТОРЫ

Цель работы: изучить возможности библиотеки jQuery, ознакомиться с понятием селектора, способами их задания и иерархией селекторов. Закрепить ранее полученные знания по HTML и JavaScript.

**ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ**

**jQuery** – это JavaScript-библиотека, фокусирующаяся на взаимодействии JavaScript, HTML и CSS.

**Возможности jQuery:**

* Обращение к любому элементу DOM (объектной модели документа) и манипулирование ими.
* Работа с событиями.
* Легкое осуществление различные визуальные эффекты.
* Работа с AJAX (технология, позволяющая общаться с сервером без перезагрузки страницы).
* Имеет огромное количество JavaScript плагинов, предназначенных для создания элементов пользовательских интерфейсов.

**Подключение библиотеки jQuery**

Сначала надо перенести библиотеку в ту же папку, где лежат html-страницы (это необязательно, но адреса для всех последующих примеров будут написаны, исходя из такой структуры). Теперь нам надо подключить jQuery к html-странице. Для этого, как вы помните, в html существует тег script, который отвечает за подключение внешних файлов скриптов.

Добавим этот тег в html-страницу:



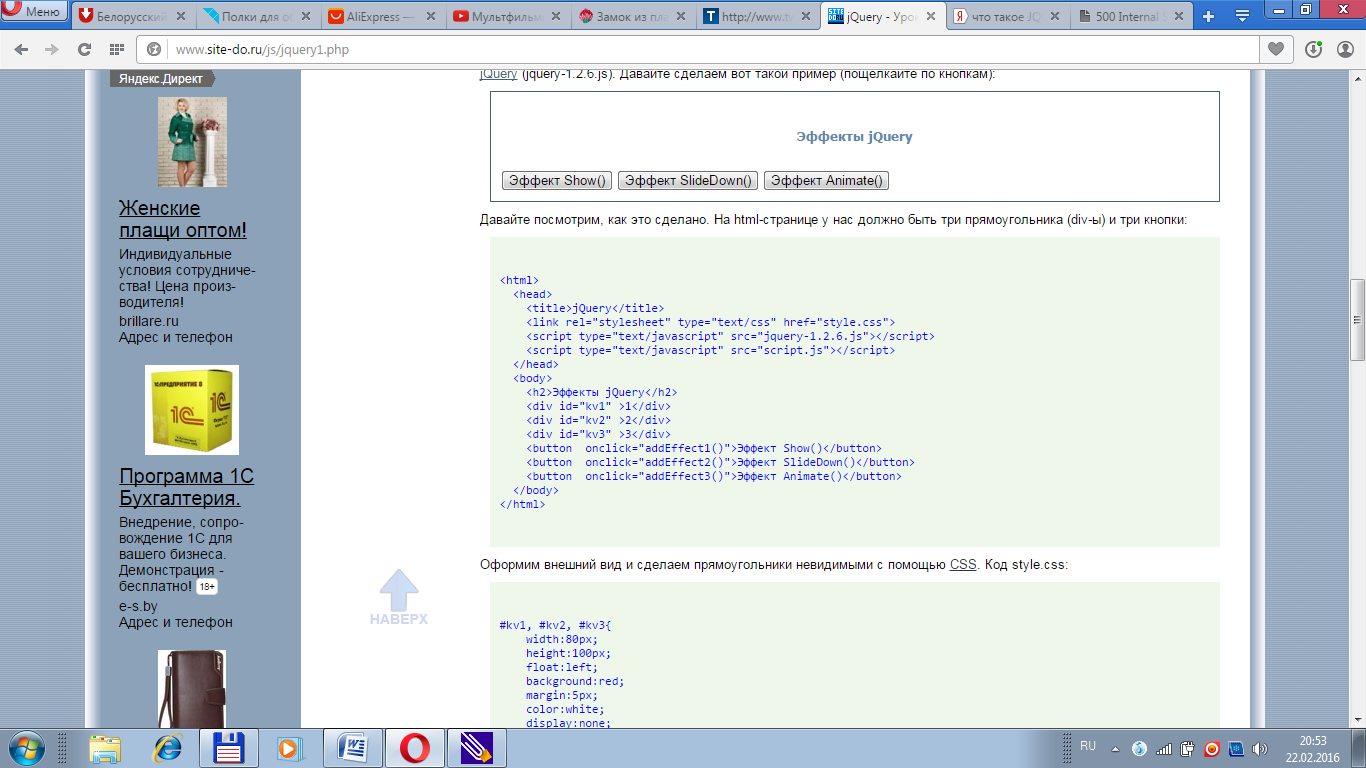
Примечание: Обратите внимание на имя файла. Здесь используется библиотека jquery-1.2.6, но Вы можете скачать более свежую версию, а потому цифры у Вас могут отличаться.

Итак, у нас в папке лежат html-страница, страница со стилями (style.css), страница с js-функциями (script.js) и библиотека jQuery (jquery-1.2.6.js).

Пример 1 (демонстрирующий работу библиотеки jQuery):

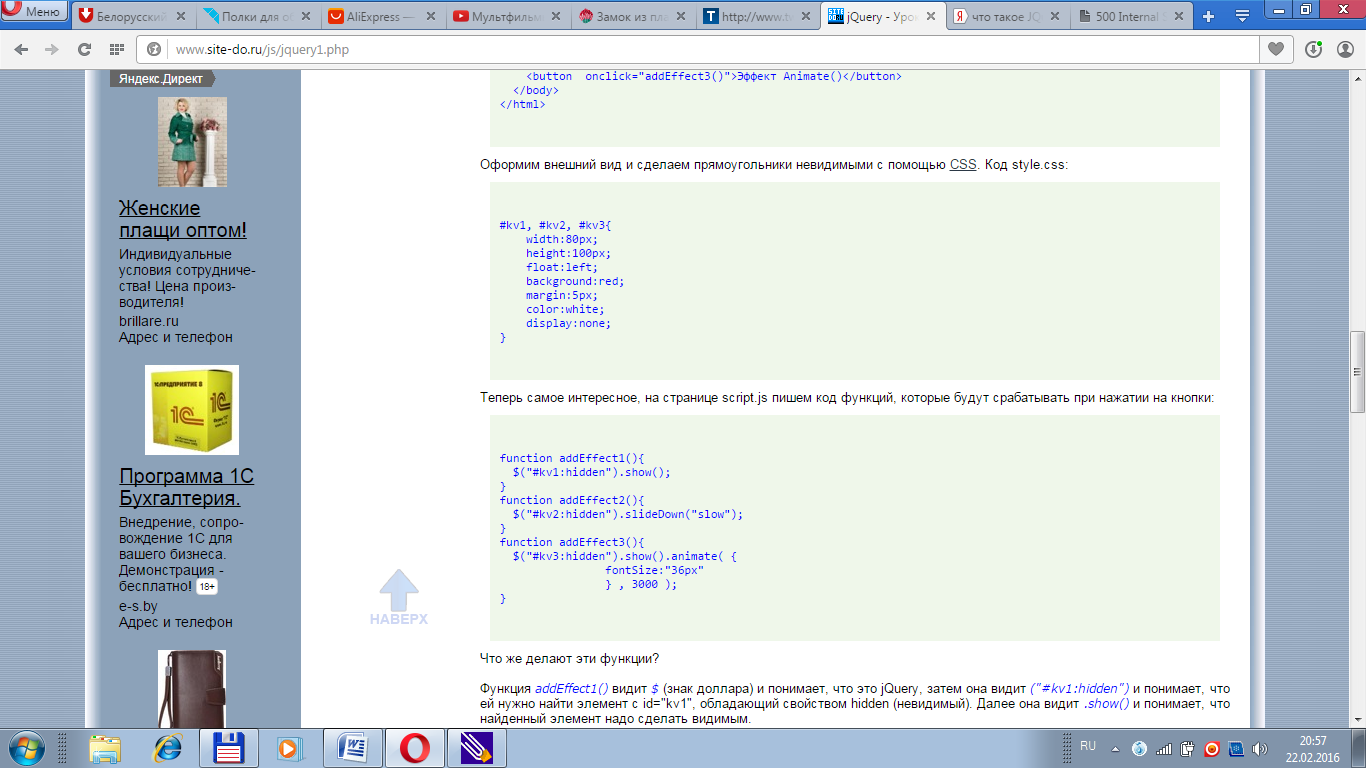


На html-странице у нас должно быть три прямоугольника (div-ы) и три кнопки: Эффект Show(), Эффект SlideDown(), Эффект Animate().



Оформим внешний вид и сделаем прямоугольники невидимыми с помощью CSS.

Код style.css



На странице script.js пишем код функций, которые будут срабатывать при нажатии на кнопки:

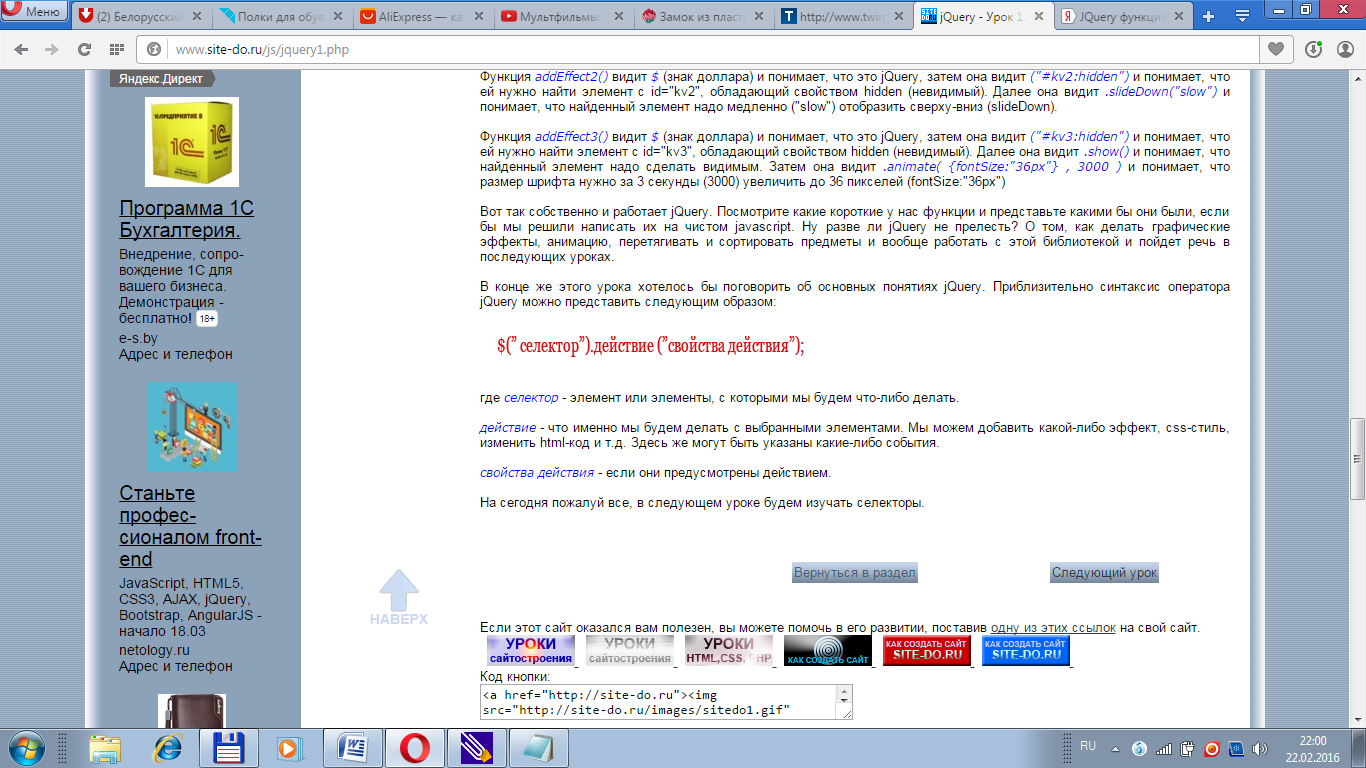


Что же делают эти функции?

Функция ***addEffect1****()* видит $ (знак доллара) и понимает, что это jQuery, затем она видит ("#kv1:hidden") и понимает, что ей нужно найти элемент с id="kv1", обладающий свойством hidden (невидимый). Далее она видит .show() и понимает, что найденный элемент надо сделать видимым.

Функция ***addEffect2****()* видит $ (знак доллара) и понимает, что это jQuery, затем она видит ("#kv2:hidden") и понимает, что ей нужно найти элемент с id="kv2", обладающий свойством hidden (невидимый). Далее она видит .slideDown("slow") и понимает, что найденный элемент надо медленно ("slow", также возможны варианты “normal”, или “fast”) отобразить сверху-вниз (slideDown).

Функция ***addEffect3****()* видит $ (знак доллара) и понимает, что это jQuery, затем она видит ("#kv3:hidden") и понимает, что ей нужно найти элемент с id="kv3", обладающий свойством hidden (невидимый). Далее она видит .show() и понимает, что найденный элемент надо сделать видимым. Затем она видит .animate( {fontSize:"36px"} , 3000 ) и понимает, что размер шрифта нужно за 3 секунды (3000) увеличить до 36 пикселей (fontSize:"36px")



Приблизительно синтаксис оператора jQuery можно представить следующим образом:

где

*селектор* - элемент или элементы, с которыми мы будем что-либо делать.

*действие* - что именно мы будем делать с выбранными элементами. Мы можем добавить какой-либо эффект, css-стиль, изменить html-код и т.д. Здесь же могут быть указаны какие-либо события,

*свойства действия* - если они предусмотрены действием.

**Селекторы**

**Селекторами** называют строчные выражения, с помощью которых задаются условия поиска элементов DOM на странице.

Давайте посмотрим, как их можно задавать:

* **#id** - выбирает единственный элемент с переданным идентификатором (id).

Пример: $("#lok").css("border", "1px solid red");

Данный оператор выберет элемент с id="lok" и добавит ему css-стиль, в данном случае рамку, шириной в 1 пиксел, сплошную и красную.

* **.class** - выбирает все элементы с переданным классом.

Пример: $(".lok").css("border", "1px solid red");

Данный оператор выберет все элементы с class="lok" и добавит им css-стиль, в данном случае рамку, шириной в 1 пиксел, сплошную и красную.

* **elements** - выбирает все элементы с выбранным именем.

Пример: $("div").css("border", "1px solid red");

Данный оператор выберет все div-ы и обведет их в рамки.

* **\*** - выбирает все элементы, включая head и body.

Пример: $("\*").css("border", "1px solid red");

Данный оператор добавит рамку ко всем элементам.

Пример: $("\*", document.body).css("border", "1px solid red");

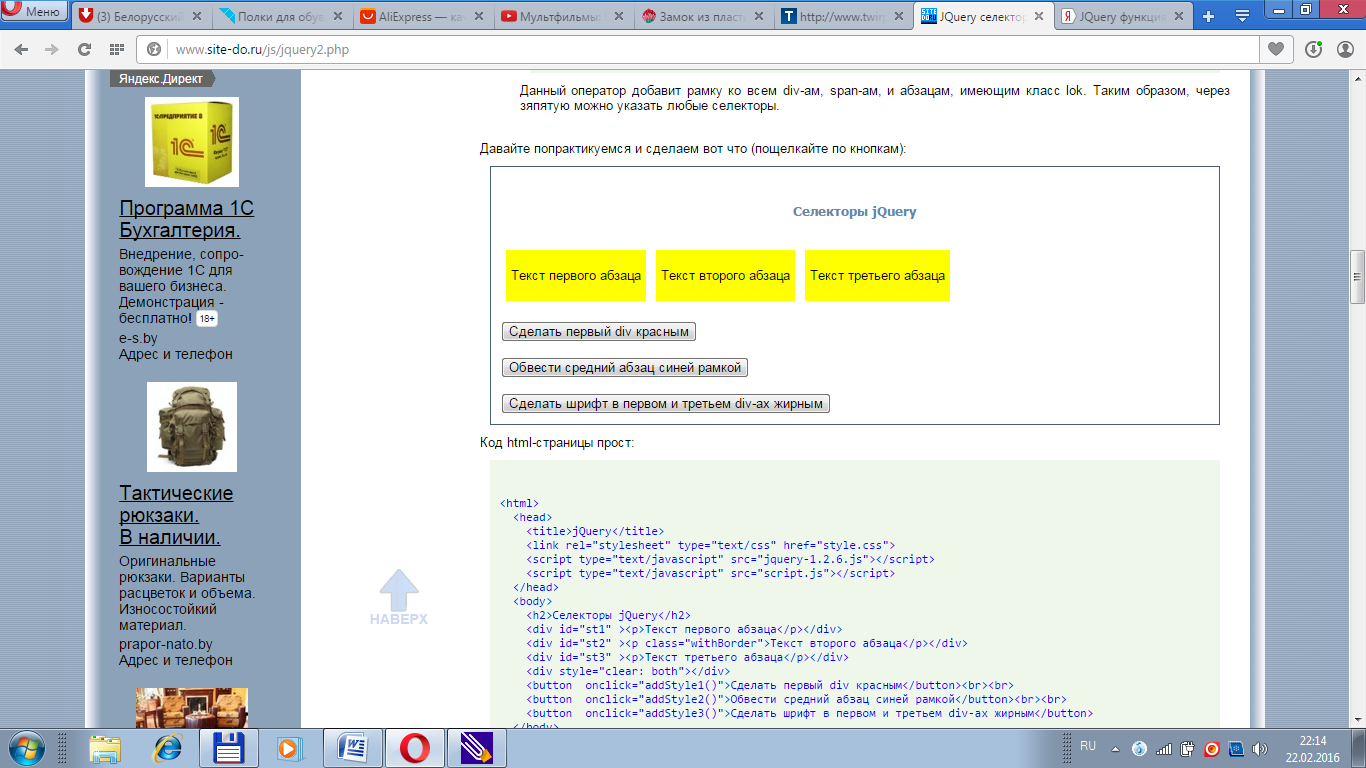
Данный оператор добавит рамку ко всем элементам в тегах body.

* **selector1, ..., selectorN** - выбирает все элементы, переданные в селекторах.

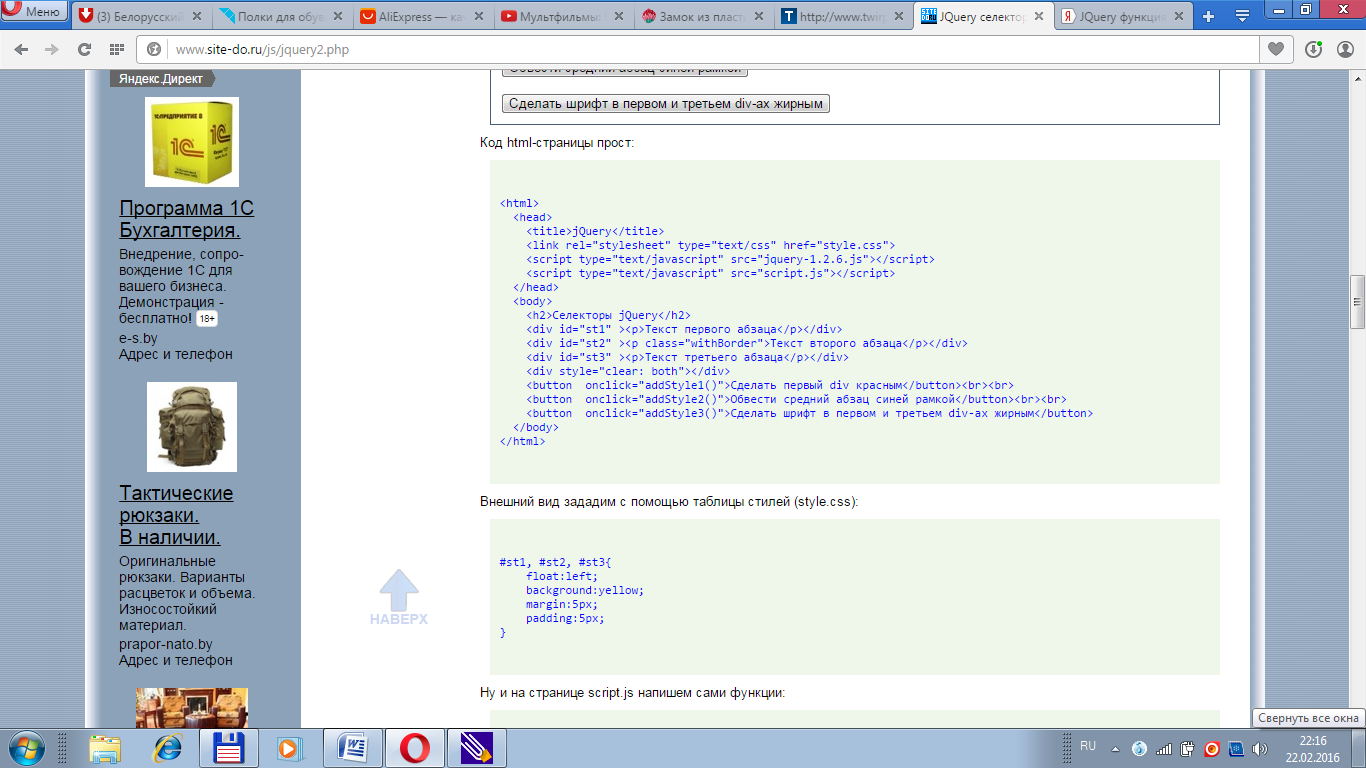
Пример: $("div, span, p.lok").css("border", "1px solid red");

Данный оператор добавит рамку ко всем div-ам, span-ам, и абзацам, имеющим класс lok. Таким образом, через запятую можно указать любые селекторы.

Пример 2: Создадим страницу следующего вида и содержания.



Код Html-страницы:



Внешний вид зададим с помощью таблицы стилей (style.css):



На странице script.js напишем сами функции



**Иерархические селекторы**

В JavaScript мы рассматривали иерархию объектов, по аналогии у элементов могут быть предки (parent) и потомки (child). Например, все элементы, находящиеся на форме являются потомками формы, а форма, в свою очередь, является их предком. А в предыдущем примере абзац с классом withBorder является потомком div-а с id="st2" (он находится внутри него).

Рассмотрим, как это можно использовать в селекторах jQuery:

* **parent child** - выбирает все элементы, являющиеся потомками для определенного элемента предка.

Пример: $("form input").css("border", "1px solid red");

Данная инструкция добавит рамку всем input-ам, которые являются потомками элемента form.

* **parent > child** - выбирает всех прямых наследников элемента-родителя.

Пример - пусть у нас есть html-страница такой структуры:



Тогда следующая инструкция: $("#content > \*").css("border", "1px solid red"); добавит рамку всем элементам (div-у, span-у и абзацу), которые находятся в div-е с id="content", но не затронет одноименные элементы в div-е с id="main".

* prev + next - выбирает все элементы, которые являются следующими (next) за элементом prev (предыдущий).

Пример: $("label + input").css("border", "1px solid red");

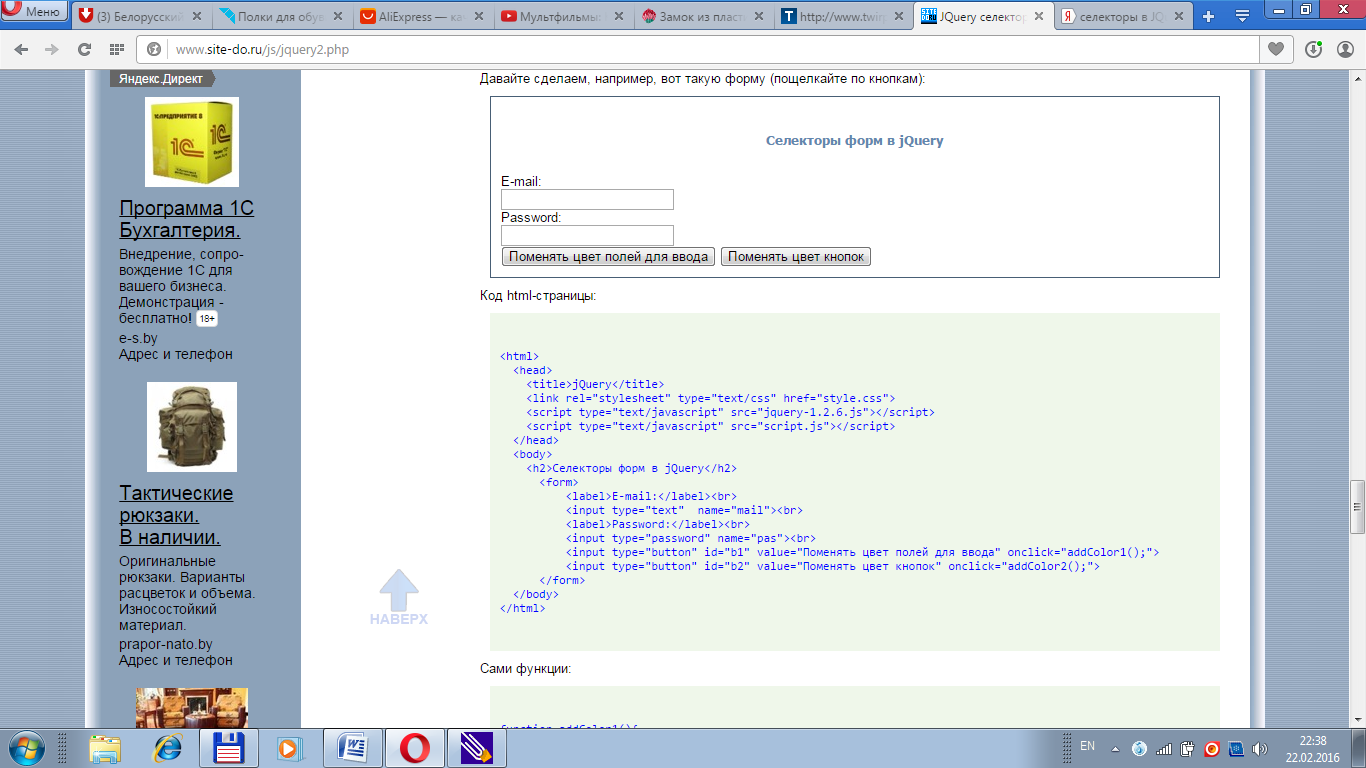
Данная инструкция добавит рамку только тем input-ам, которые следуют непосредственно за надписями (label).

**Селекторы форм**

Перечислим селекторы формы

* :input - выбираются все элементы input.
* :text - выбираются все элементы input типа text. :
* :password - выбираются все элементы input типа password.
* :radio - выбираются все элементы input типа radio.
* :checkbox - выбираются все элементы input типа checkbox.
* :submit - выбираются все элементы input типа submit.
* :reset - выбираются все элементы input типа reset.
* :button - выбираются все элементы input типа button.
* :image - выбираются все элементы input типа image.
* :file - выбираются все элементы input типа file.
* :hidden - выбираются все элементы input типа hidden или просто скрытые.

Пример 3. Создадим следующую форму:



Код html-страницы (напишите самостоятельно, задавая id=b1 для кнопки с названием «Поменять цвет полей для ввода» и id=b2 для кнопки с названием «Поменять цвет кнопок»):

Сами функции



Задание. Поменять цвет только у одной кнопки, например, у первой, обращаясь к ней по ее id.

Т.е. селекторы форм помогают совершать массовые действия с элементами формы.

**Контрольные вопросы:**

1. Что представляет собой jQuery?
2. Перечислите возможности библиотеки jQuery.
3. Как производится подключение библиотеки jQuery к html-странице?
4. Какой синтаксис имеет оператор jQuery?
5. Что такое селектор?
6. Как можно задавать селекторы?
7. Как можно использовать иерархию объектов в селекторах jQuery?
8. Перечислите селекторы форм.

**Задания:**

1. Изучите теоретические сведения.
2. Изучите работу 3-ех примеров на практике (выполните имеющиеся задания)
3. Изучите приложение.
4. Оформите отчет о лабораторной работе.

**Содержание отчета:**

Лабораторная работа № 1

Тема лабораторной работы.

Цель работы.

Краткие теоретические сведения (ответы на контрольные вопросы).

Вывод о проделанной работе.

Приложение

