# Лабораторная работа № 3

## МЕТОДЫ ДЛЯ РАБОТЫ С CSS-СТИЛЯМИ.

## МЕТОДЫ ДЛЯ РАБОТЫ С HTML

Цель работы: ознакомиться с методами работы CSS-СТИЛЯМИ и HTML, приобрести практические умения написания кода с использованием некоторых методов.

**ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ**

**МЕТОДЫ ДЛЯ РАБОТЫ С CSS-СТИЛЯМИ**

Рассмотрим методы, которые используются для CSS-стилей. Итак, чтобы добавить какому-либо элементу стиль, необходимо воспользоваться следующим методом:

* .**css(name,value)**

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

Данная инструкция обведет div синей рамкой. В качестве параметров здесь используются названия и значения, применимые в CSS: background, border, font-style, color и т.д. Если необходимо задать для элемента несколько CSS-правил, то лучше использовать следующую конструкцию:

* .**css({properties})**

Пример: $("div").css({

border:"1px solid blue",

fontWeight:"bolder",

backgroundColor:"red"

});

Данная инструкция обведет div синей рамкой, сделает фон красным, а текст - жирным. Обратите внимание, что для сложносоставных свойств CSS вроде font-weight и background-color используются иэ эквиваленты из JS: fontWeight, backgroundColor и т.д.

Перечислим другие методы для работы со стилями:

* **.addClass(class)**

Пример: $("p:last").addClass("main");

Данная инструкция добавит класс main к последнему элементу параграфа.

* **.removeClass(class)**

Пример: $("p:even").removeClass("main");

Данная инструкция удалит класс main из всех четных параграфов.

* **.toggleClass(class)**

Пример: $("p").toggleClass("main");

Данная инструкция удалит класс main из всех параграфов, если он присутствует. И добавит этот класс, если он отсутствует.

* .**offset()**

Пример: var offDiv=$("div").offset();

Данная инструкция позволяет получить отступы слева и сверху для элемента. Чтобы получить значения конкретного свойства, нужно использовать следующие свойства: offset.left для отступа слева и offset.top - для отступа сверху.

* .**height(value)**

Пример: $("div").height();

$("div").height(200);

Данная инструкция позволяет получить (первая строка) и задать (вторая строка) высоту элемента.

* .**width(value)**

Пример: $("div").width();

$("div").width(200);

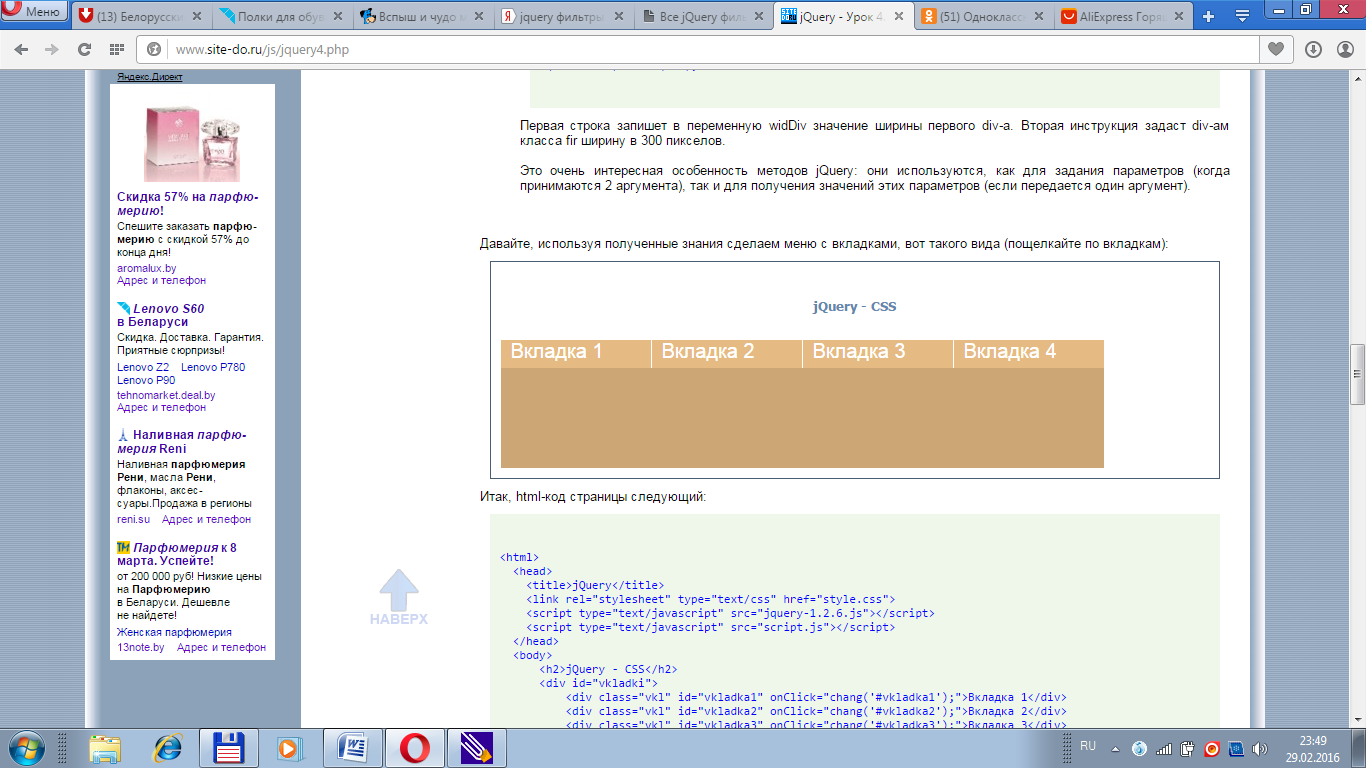
Данная инструкция позволяет получить (первая строка) и задать (вторая строка) ширину элемента.

Еще пример: var widDiv=$("div").width();

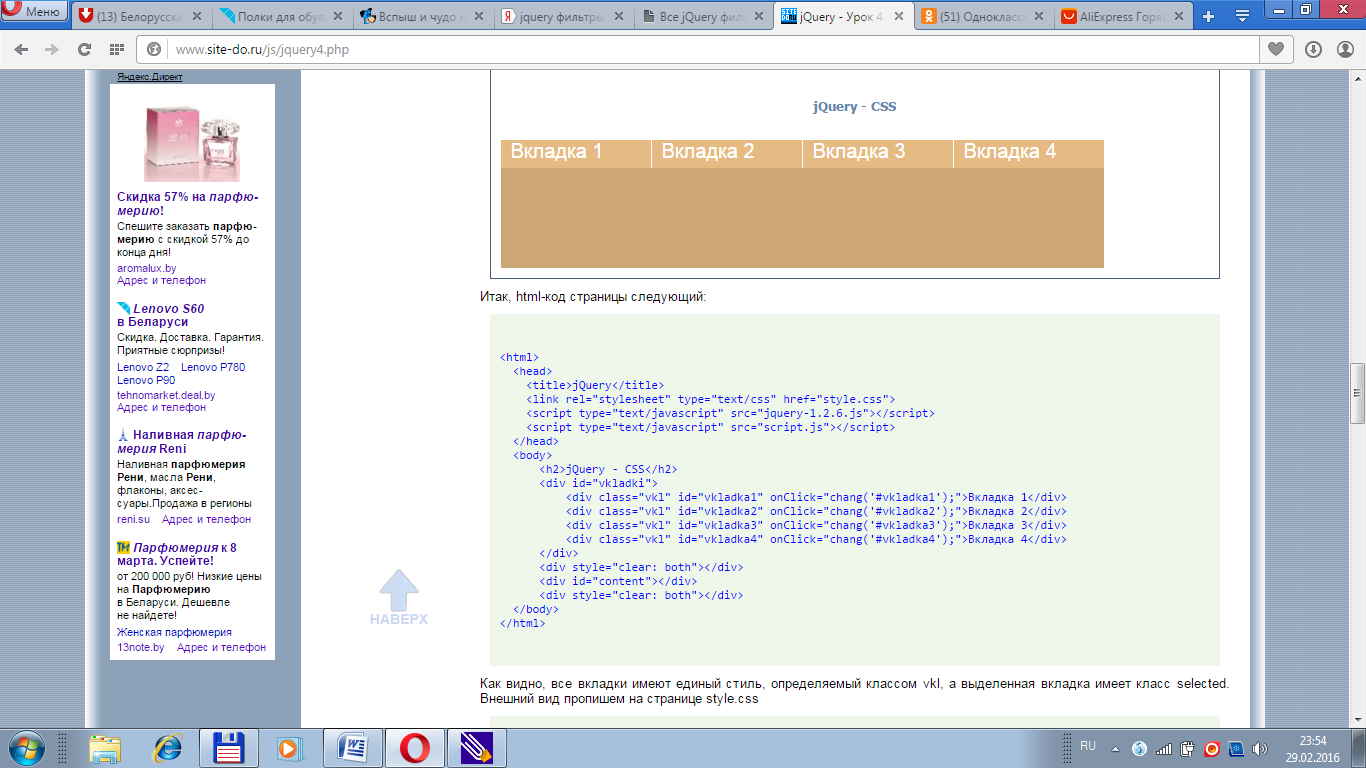
$("div.fir").width(300);

Первая строка запишет в переменную widDiv значение ширины первого div-а. Вторая инструкция задаст div-ам класса fir ширину в 300 пикселов. Это очень интересная особенность методов jQuery: они используются, как для задания параметров (когда принимаются 2 аргумента), так и для получения значений этих параметров (если передается один аргумент).

Пример 1: создадим меню с вкладками, вот такого вида



HTML-код выглядит так:



Как видно, все вкладки имеют единый стиль, определяемый классом vkl, а выделенная вкладка имеет класс selected. Внешний вид пропишем на странице style.css

.vkl{ float:left;

width:140px;

height:28px;

background:#E5BA83;

color:white;

font-size:20px;

border-right:1px solid white;

padding-left:10px;

}

#content{

width:603px;

height:100px;

background:#CCA675;

}

.selected

{

background:#CCA675;

}

Функция chang():

function chang(s){

$('.selected').removeClass('selected');

$(s).addClass('selected');

}

Функция работает следующим образом: сначала ищется элемент, который имеет класс selected, и этот класс у него убирается. Затем (вторая строка) выделенному элементу (он находится по id, переданному в функцию в качестве параметра s) добавляется класс selected.

## Методы для работы с HTML

Основные методыработы с HTML:

* **html(val)** - добавляет html-код в выбранные элементы.

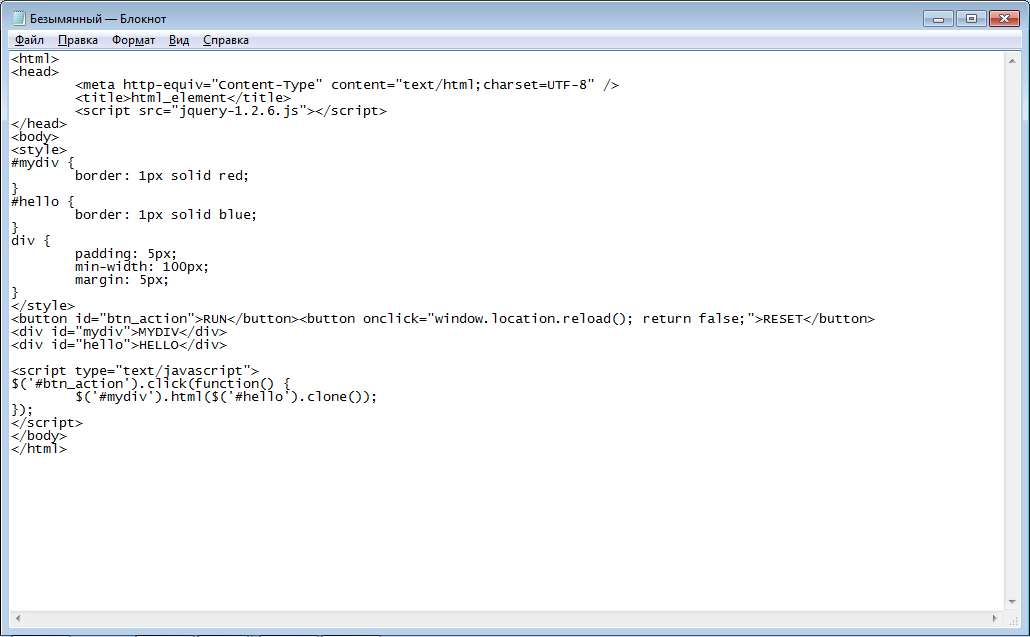
Пример: $("div.sp").html("Привет");

Данная инструкция добавит во все div-ы с классом sp, span-ы с текстом "Привет".

Пример 2. Поместить текст «Hello» в синей рамке, в красную рамку.

|  |  |
| --- | --- |
| Исходный вид | Результат |
|  |  |

HTML-код



* **html(function(index, value)) -**

Заменяет содержимое выбранных элементов на возвращенное пользовательской функцией значение.

Функция вызывается отдельно, для каждого из выбранных элементов. При вызове ей передаются следующие параметры:

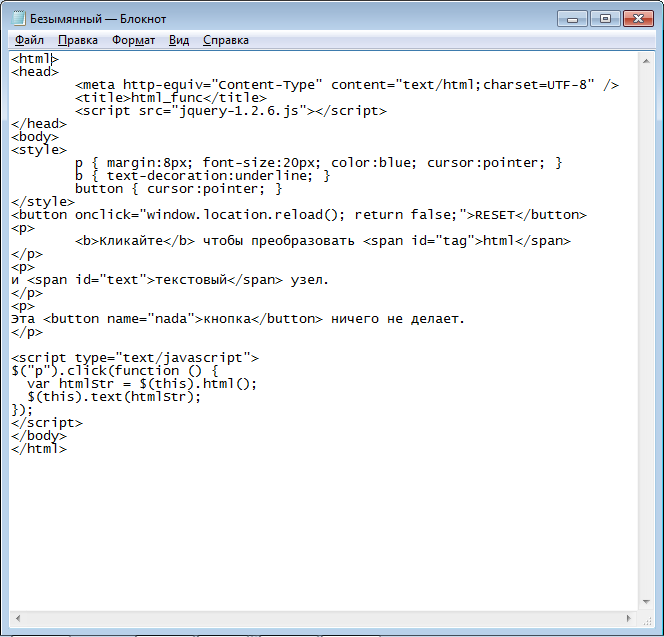
**index** - позиция элемента в наборе,

**value** - текущее html-содержимое.

Пример 3: При щелчке мышкой на параграфе конвертировать его HTML-код в текст.

|  |  |
| --- | --- |
| Исходный вид | Результат |
|  |  |

HTML-код

****

* **text() / val()** - возвращает текстовое содержимое элемента / значение элемента.

Пример: $("p").text();

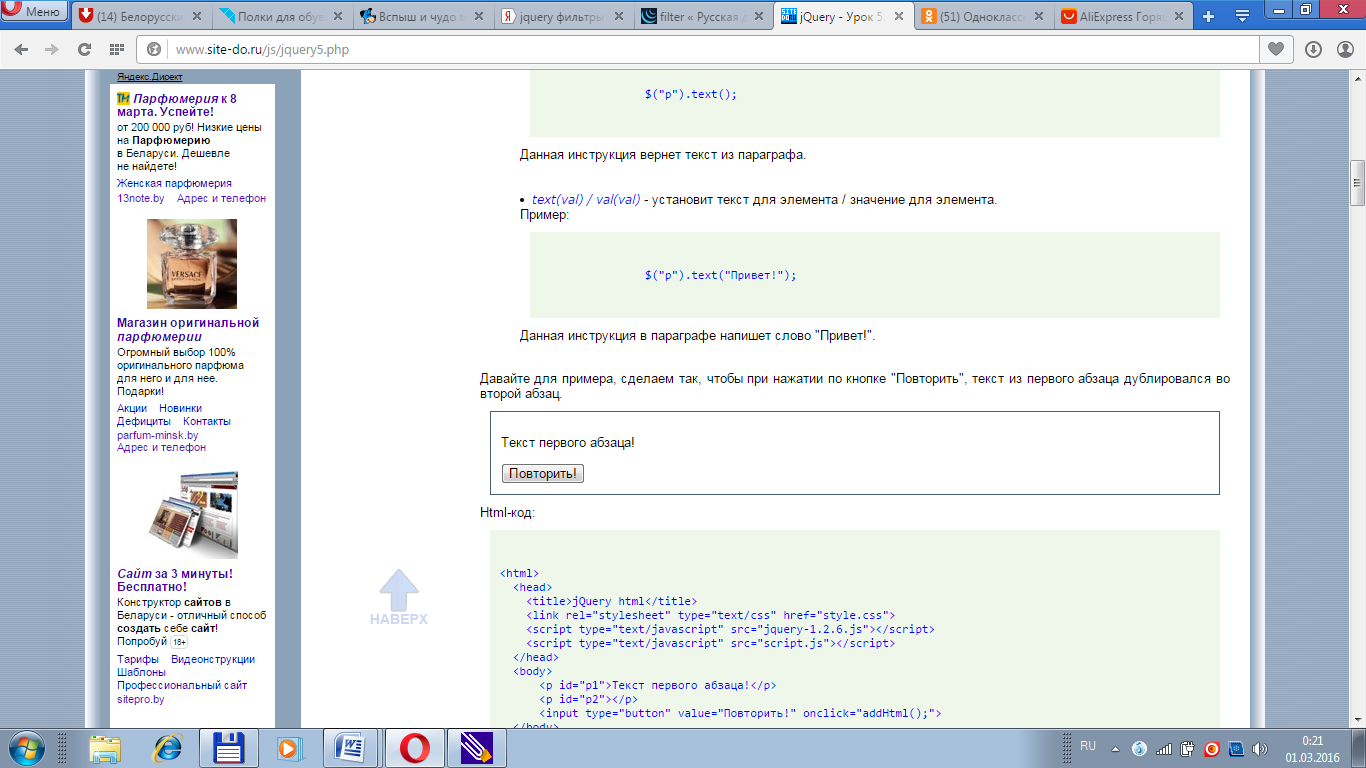
Данная инструкция вернет текст из параграфа.

* **text(val) / val(val)** - установит текст для элемента / значение для элемента.

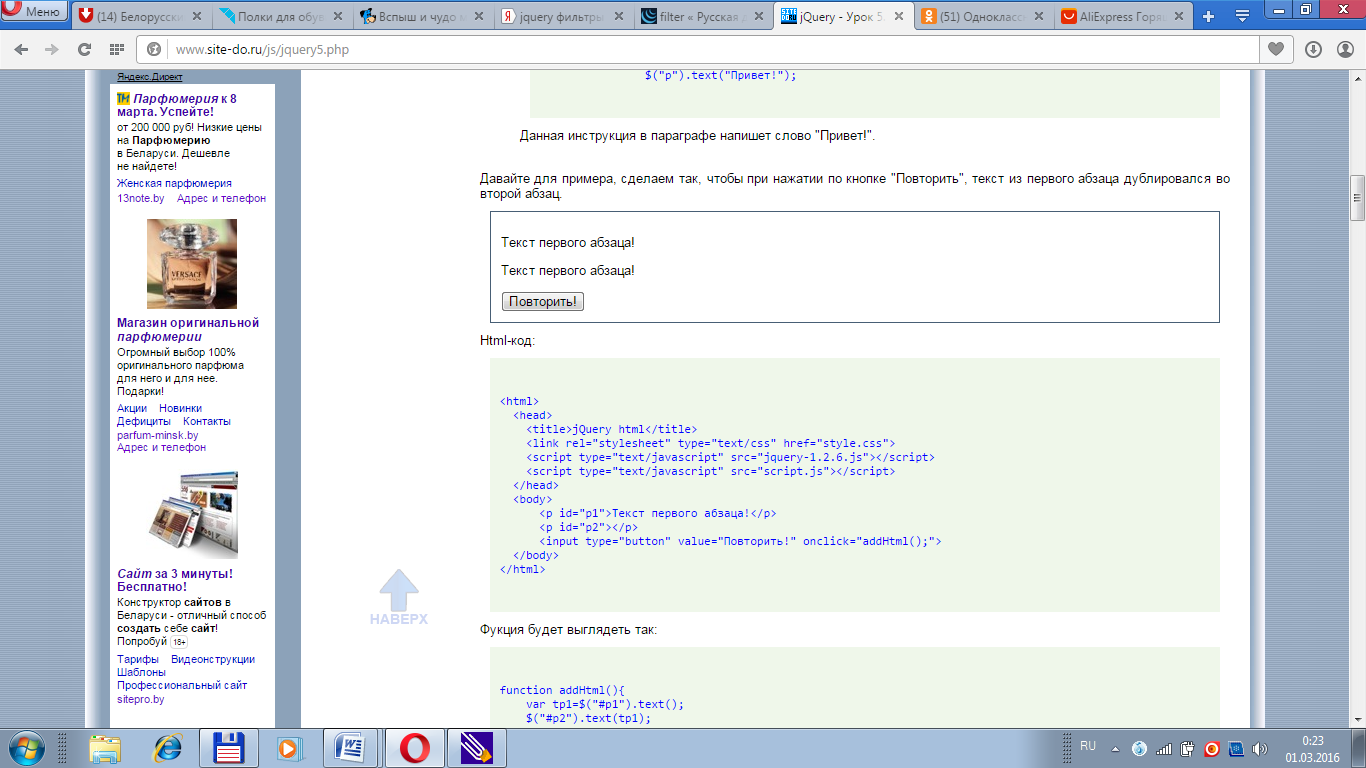
Пример: $("p").text("Привет!");

Данная инструкция в параграфе напишет слово "Привет!".

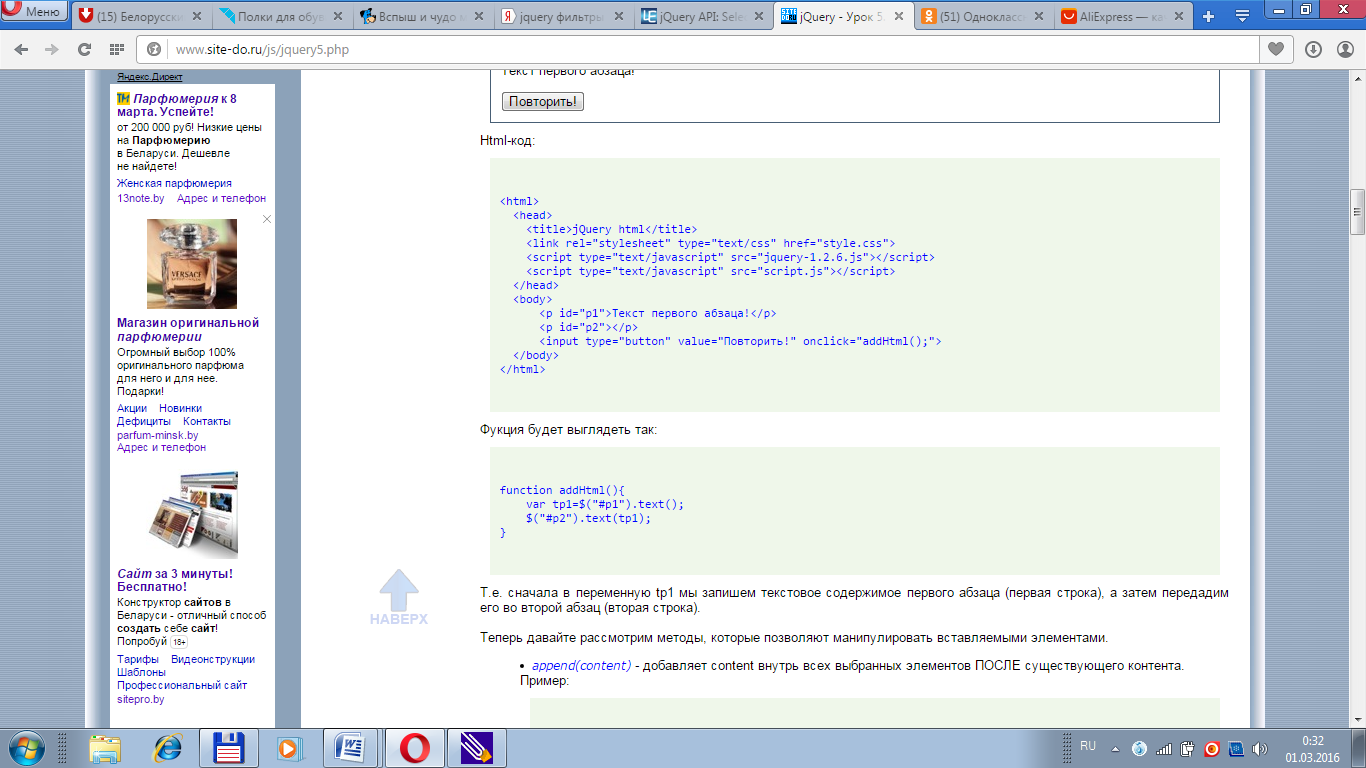
Пример 4: Сделать так, чтобы при нажатии по кнопке "Повторить", текст из первого абзаца дублировался во второй абзац.



Html-код:



 Фукция будет выглядеть так:



Т.е. сначала в переменную tp1 мы запишем текстовое содержимое первого абзаца (первая строка), а затем передадим его во второй абзац (вторая строка).

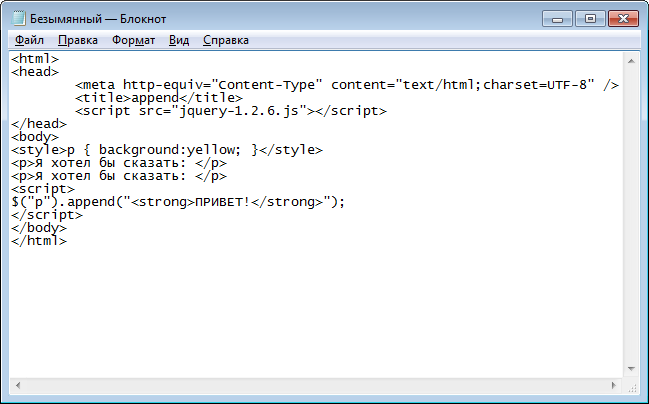
## Методы, которые позволяют манипулировать вставляемыми элементами.

* **append(content)** - добавляет content внутрь всех выбранных элементов ПОСЛЕ существующего контента.

Пример: $("p").append("Привет");

Данная инструкция добавит в конец параграфа текст "Привет" жирным шрифтом.

Пример 5: Добавить текст «Привет!» в конец всех параграфов.



* **appendTo(content)** - обратная операция, добавляет в content все выбранные элементы.

Пример: $("p").appendTo("#main");

Данная инструкция добавит параграф в конец элемента с идентификатором "main". Иными словами: $(A).append(B) - добавит В в конец А. $(A).appendTo(B) - добавит А в конец В.

* **prepend(content)** - добавляет content внутрь всех выбранных элементов ДО существующего контента.

Пример: $("p").prepend("Привет");

Данная инструкция вставит в начало параграфа текст "Привет" жирным шрифтом.

Задание 1: Добавить текст «Привет!» в начало всех параграфов.

* **prependTo(content)** - обратная операция, добавляет В content все выбранные элементы.

Пример: $("p").prependTo("#main");

Данная инструкция вставит параграф в начало элемента с идентификатором "main". Иными словами: $(A).prepend(B) - добавит В в начало А. $(A).prependTo(B) - добавит А в начало В.

* **after(content)** - добавляет content ПОСЛЕ всех выбранных элементов (заметьте ПОСЛЕ элемента, а не в конец элемента, как в случае с append.

Пример: $("p").after("Привет");

Данная инструкция вставит после параграфа текст "Привет" жирным шрифтом.

* **befor(content)** - добавляет content ДО всех выбранных элементов (заметьте ДО элемента, а не в начало элемента, как в случае с prepend.

Пример: $("p").befor("Привет");

Данная инструкция вставит перед параграфом текст "Привет" жирным шрифтом.

* **insertAfter(content)** - вставляет все выбранные элементы ПОСЛЕ content-а.

Пример: $("p").insertAfter("#main");

Данная инструкция вставит параграф после элемента с идентификатором "main". Иными словами: $(A).after(B) - вставит В после А. $(A).insertAfter(B) - вставит А после В.

* **insertBefore(content)** - вставляет все выбранные элементы ПЕРЕД content-ом .

Пример: $("p").insertBefore("#main");

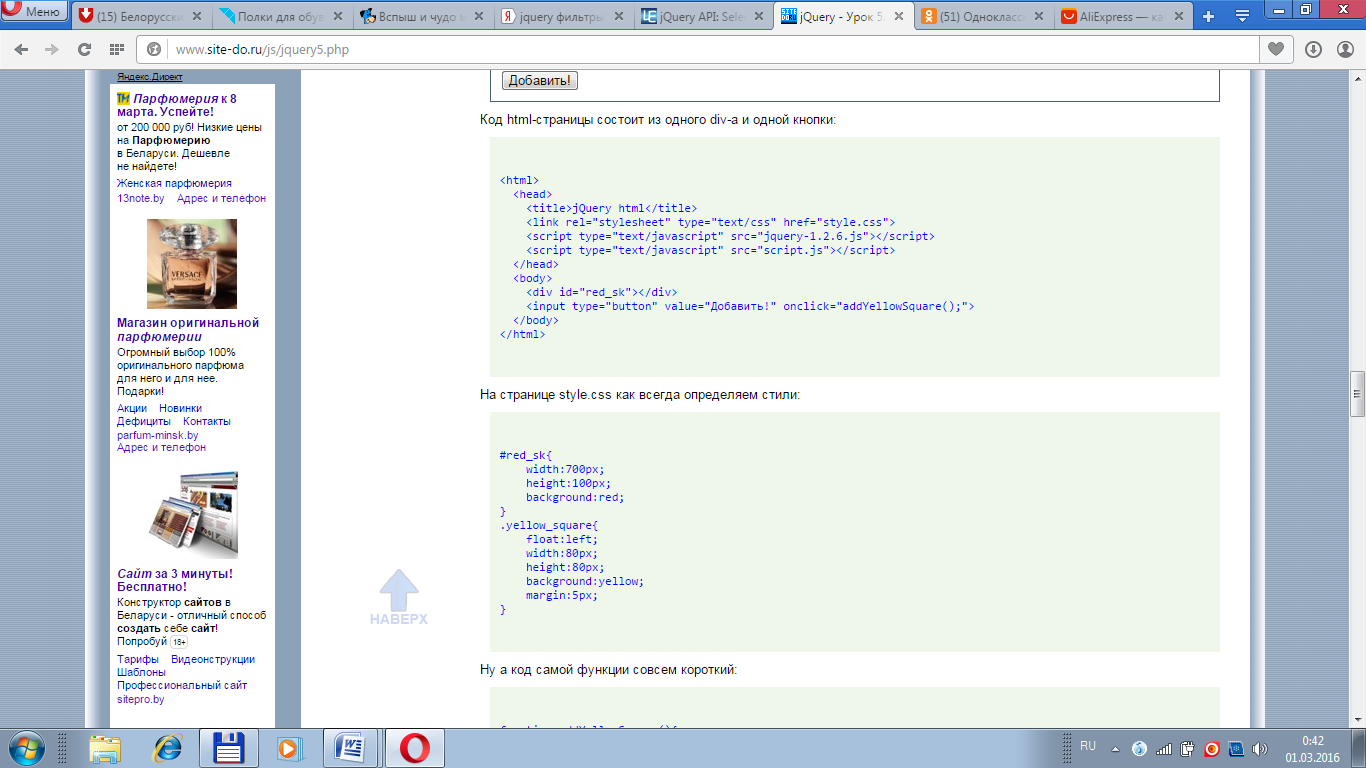
Данная инструкция вставит параграф перед элементом с идентификатором "main".

Иными словами: $(A).before(B) - вставит В перед А. $(A).insertBefore(B) - вставит А перед В.

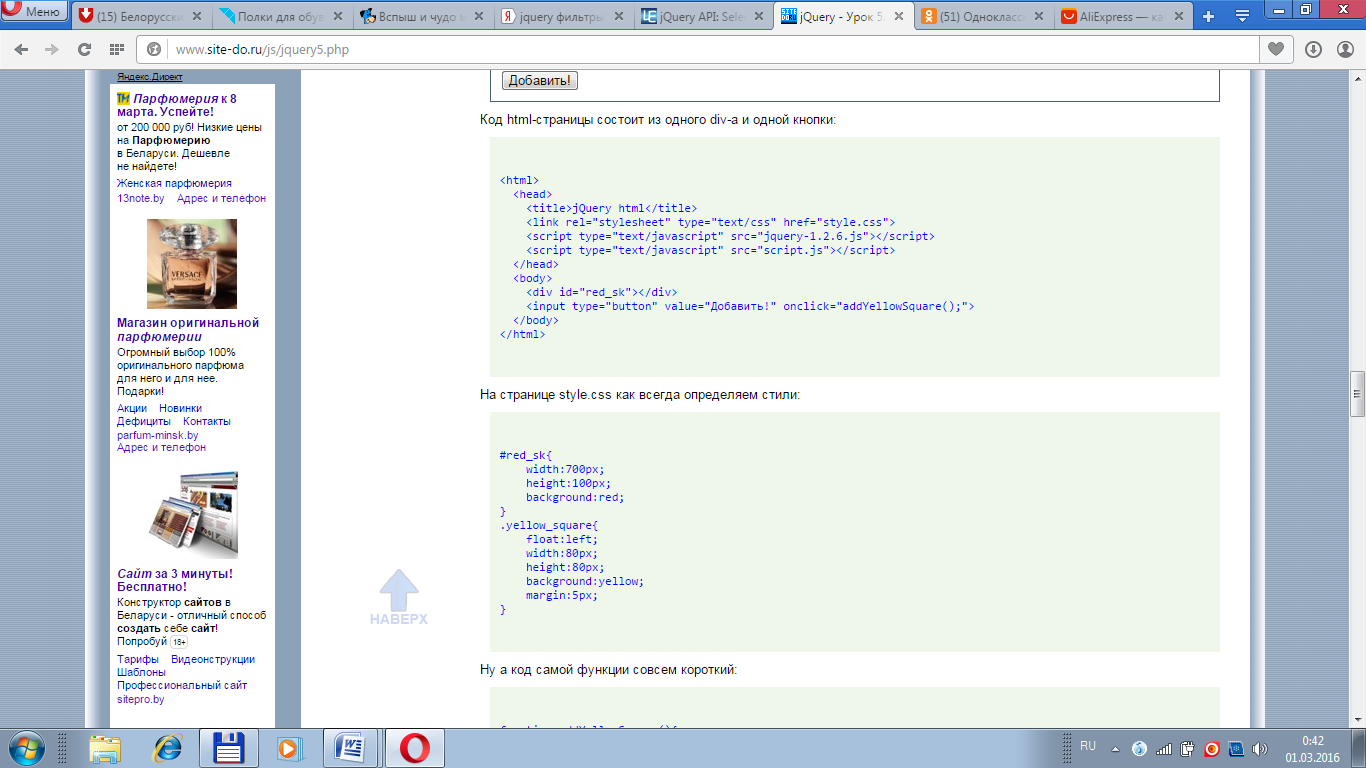
Пример 6. Сделаем так, чтобы при нажатии на кнопку "Добавить" в красном прямоугольнике появлялись желтые квадратики (один клик - один квадратик).



Код html-страницы состоит из одного div-а и одной кнопки:



 На странице style.css как всегда определяем стили:



Код самой функции выглядит так:



Задание 2: Добавьте 5 желтых прямоугольников на страницу. При нажатии на прямоугольник должен изменяться его цвет и высота становиться 30 пикселей. При нажатии на кнопку **Reset** страница принимает первоначальный вид.

Задание 3: Добавьте 5 желтых прямоугольников на страницу. При нажатии на прямоугольник должен изменяться его цвет и ширина становиться 30 пикселей. При нажатии на кнопку **Reset** страница принимает первоначальный вид.

Продолжим рассматривать остальные методы:

* **wrap(html)** - оборачивает каждый элемент в элемент обертку.

Пример: $("p").wrap("<span> </span>" );

Данная инструкция обернет каждый параграф span-ом.

* **wrapAll(html)** - оборачивает все элементы в ОДИН элемент обертку.

Пример: $("p").wrapAll("<span> </span>");

Данная инструкция обернет все параграфы одним span-ом.

* **wrapInner(html)** - оборачивает внутренне содержание каждого элемента.

Пример: $("p").wrapInner("<em> </em>");

Данная инструкция выделит содержимое каждого параграфа курсивом.

* **replaceWith(content)** - замещает одни элементы другими.

Пример: $("span").replaceWith("<div> </div>");

Данная инструкция заменит все span-ы div-ами.

* **replaceAll(selector)** - обратная операция, т.е. все selector-ы будут поменяны на элементы.

Пример: $("span").replaceAll("<div> </div>");

Данная инструкция заменит все div-ы span-ами.

* **empty()** - удаляет из элемента все узлы-потомки, включая текст.

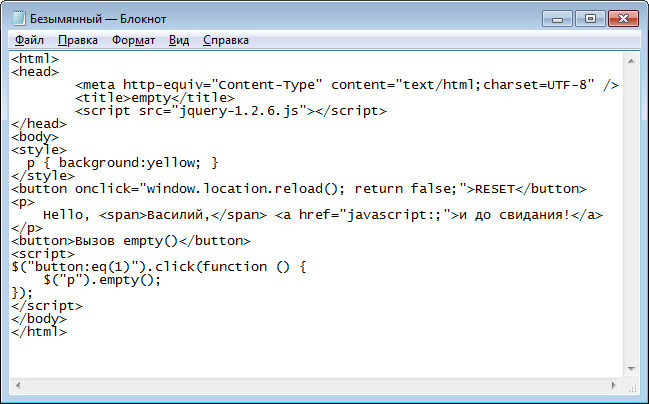
Пример: $("span").empty();

Данная инструкция удалит все содержимое span-ов.

Пример 7: Удалить все узлы-потомки из всех параграфов.

|  |  |
| --- | --- |
| Исходный вид | Результат |
|  |  |

HTML-код



* **remove()** - удаляет сами элементы.

Пример: $("span").remove();

Данная инструкция удалит все span-ы.

Задание 4: Удалить все параграфы из документа

|  |  |
| --- | --- |
| Исходный вид | Результат |
|  |  |

* **clone()** - клонирует элементы.

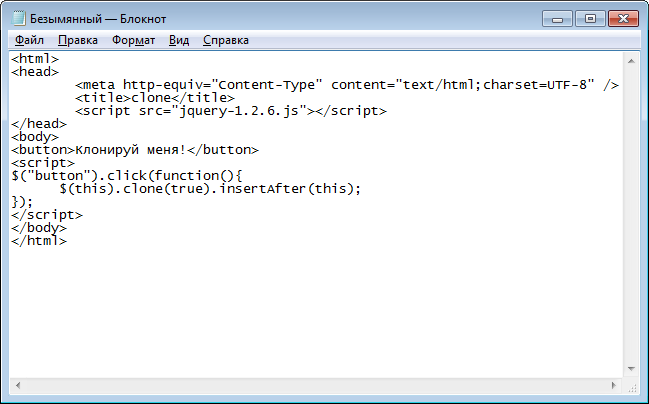
Пример: $("b").clone().append("div");

Данная инструкция возьмет элементы, выделенные жирным и добавит их в div.

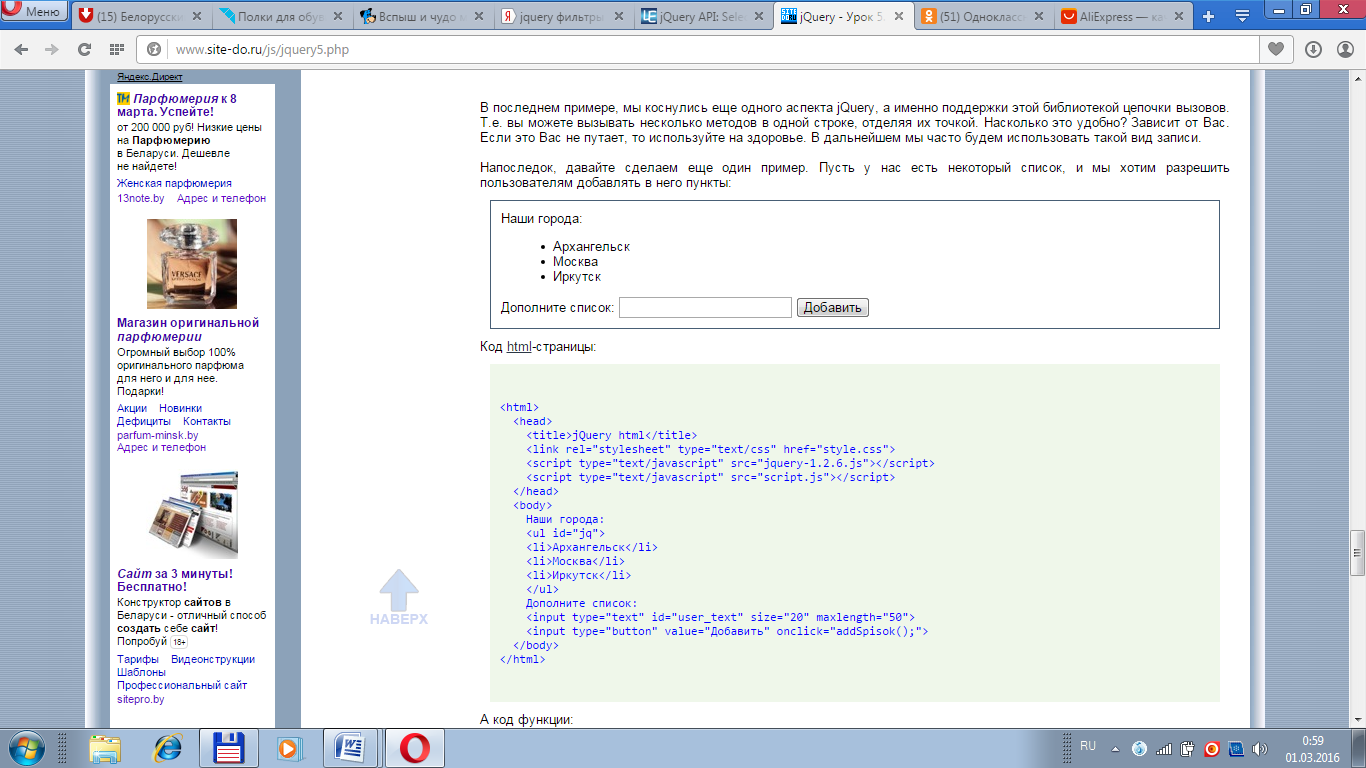
В примере, мы коснулись еще одного аспекта jQuery, а именно поддержки этой библиотекой цепочки вызовов. Т.е. вы можете вызывать несколько методов в одной строке, отделяя их точкой.

Пример 8. Создать кнопку, которая может клонировать сама себя и иметь клонов, также способных себя же клонировать.

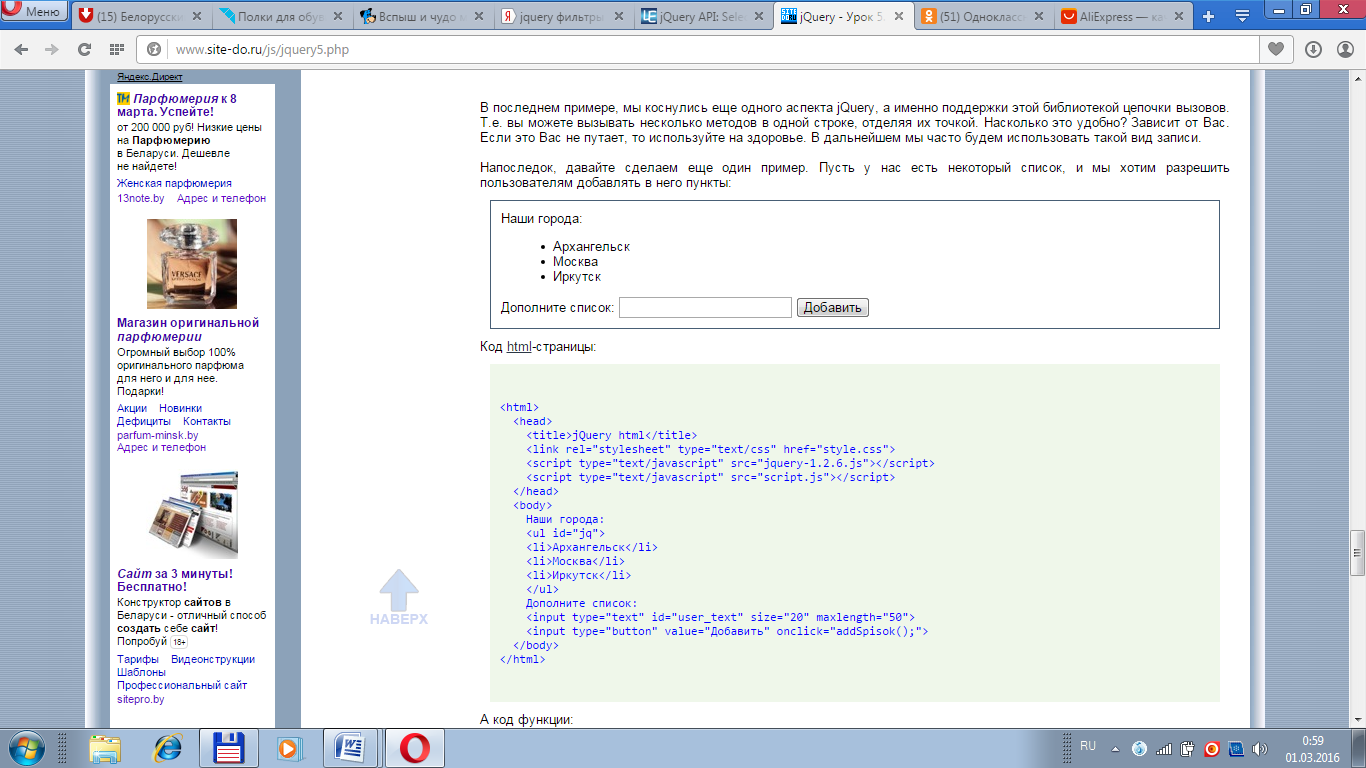
HTML-код



Пример 9. Пусть у нас есть некоторый список, и мы хотим разрешить пользователям добавлять в него пункты:



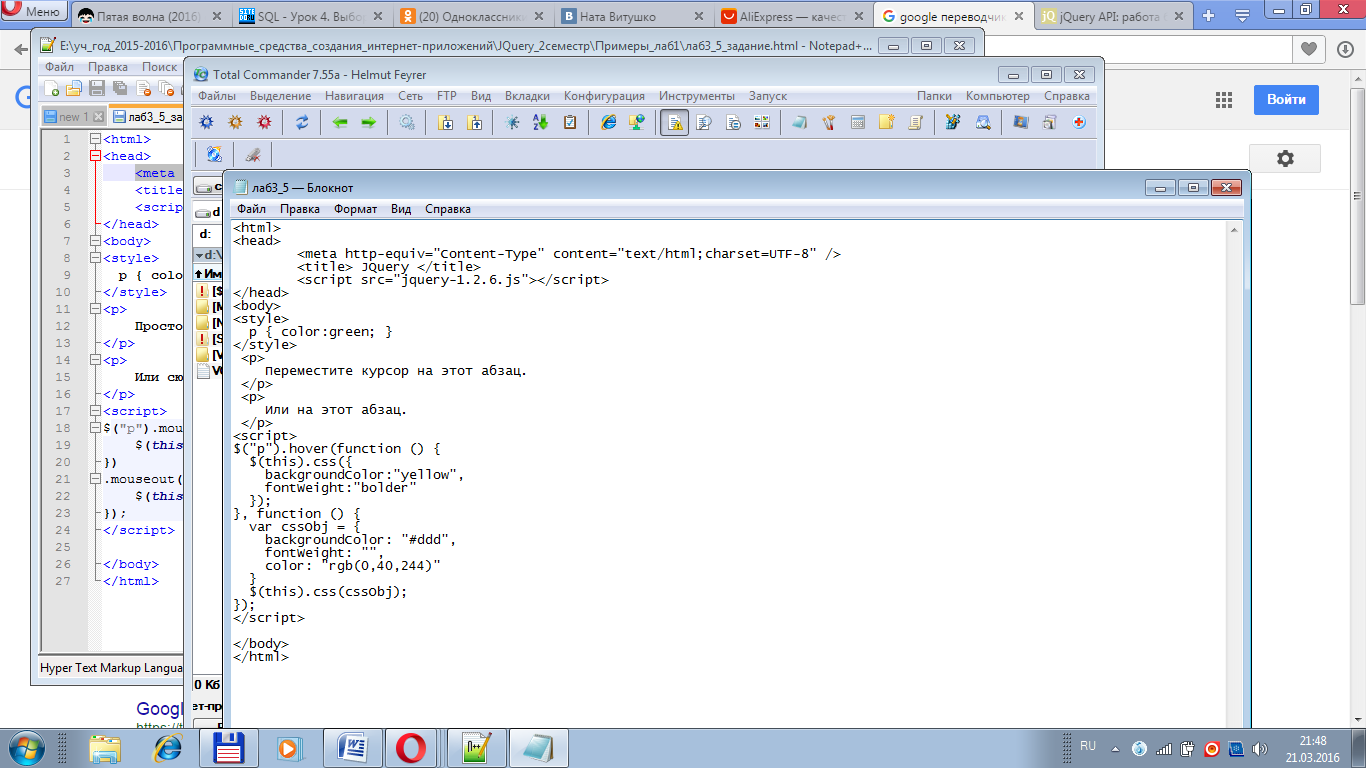
Код html-страницы:



Код функции:



Пример 10: Если указатель мыши проходит над параграфом, то для этого параграфа будет установлен желтый фон и жирное начертание текста. Когда же указатель мыши выходит за пределы параграфа - фон изменяется на серый, начертание текста становится нормальным.



Задание 5: для того чтобы изменить цвет текста любого параграфа, используйте события **mouseover**, **mouseout**.

($("p").mouseover(function () {

    $(this).css("background-color",'red');

})

.mouseout(function(){

    $(this).css("background-color",'#87CEFA');

});

**Контрольные вопросы:**

1. Перечислите методы, используемые для CSS-стилей. Каково их назначение?
2. Перечислите методы для работы с HTML. Каково их назначение?
3. Перечислите методы, которые позволяют манипулировать вставляемыми элементами. Каково их назначение?

**Задания:**

1. Изучите теоретические сведения.
2. Изучите работу примеров на практике. Выполните задания, сопровождающие примеры.
3. Оформите отчет о лабораторной работе.
4. Нарисуйте шахматную доску (8 х 8 клеток)

**Содержание отчета:**

Лабораторная работа № 3

Тема лабораторной работы.

Цель работы.

Краткие теоретические сведения (ответы на контрольные вопросы).

Вывод о проделанной работе.