# Лабораторная работа № 6

## ЭФФЕКТЫ, АНИМАЦИЯ

Цель работы: ознакомиться с методами видимости, свертывания, исчезновения, анимации.

**ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ**

jQuery имеет ряд функций, выполняющих анимационные эффекты с элементами страницы. Начиная от простых возможностей: скрытия, появления, перемещения элементов, библиотека позволит создавать свои, более сложные эффекты, основанные на изменении CSS свойств.

Эти изменения могут происходить плавно или мгновенно, замедляться, ускоряться или выполняться равномерно.

**Методы видимости**

* **.show() .hide()**

Мгновенно показывает/скрывает выбранные элементы, установив их css-свойство display в none, не изменяя при этом их прозрачность и размеры.

Синтаксис: .show(duration,[callback]) .hide(duration,[callback])

**duration** — продолжительность выполнения анимации (появления или скрытия).

Может быть задана в миллисекундах (200 и 600 миллисекунд). Может быть задана строковым значением '**fast**' или '**slow**'. Если этот параметр не задан, анимация будет происходить мгновенно, элемент просто появится/исчезнет

**callback** — функция, заданная в качестве обработчика завершения анимации (появления или скрытия).

Синтаксис:

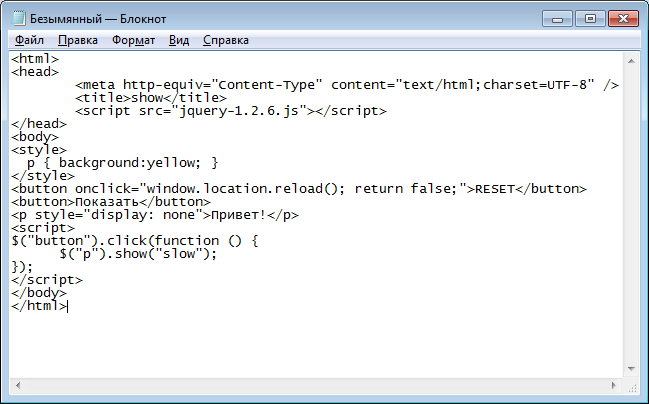
**.**show([duration],[easing],[callback]).hide([duration],[easing],[callback])

**easing** — изменение скорости анимации (будет ли она замедляется к концу выполнения или наоборот ускорится).

jQuery доступны лишь две такие функции: 'linear' и 'swing'  (для равномерной анимации и анимации с ускорением).

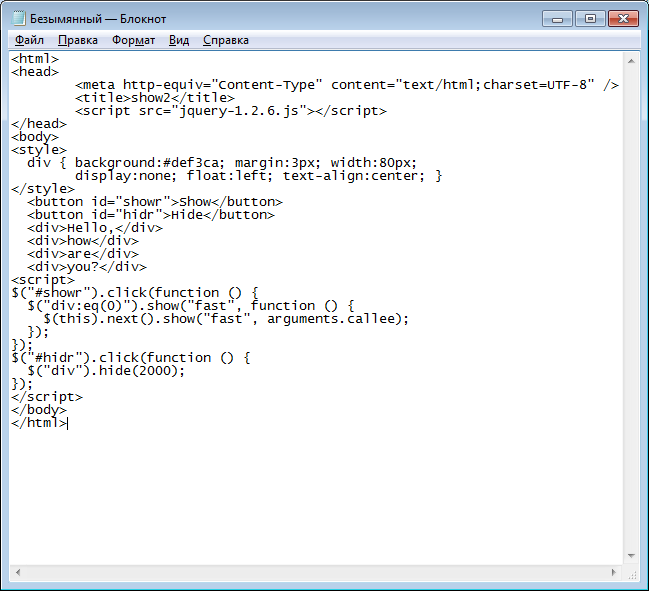
Пример 1: Показать все скрытые параграфы. Использовать в качестве параметра ключевое слово slow. При этом анимация выполняется в течение 600 миллисекунд.

HTML-код:

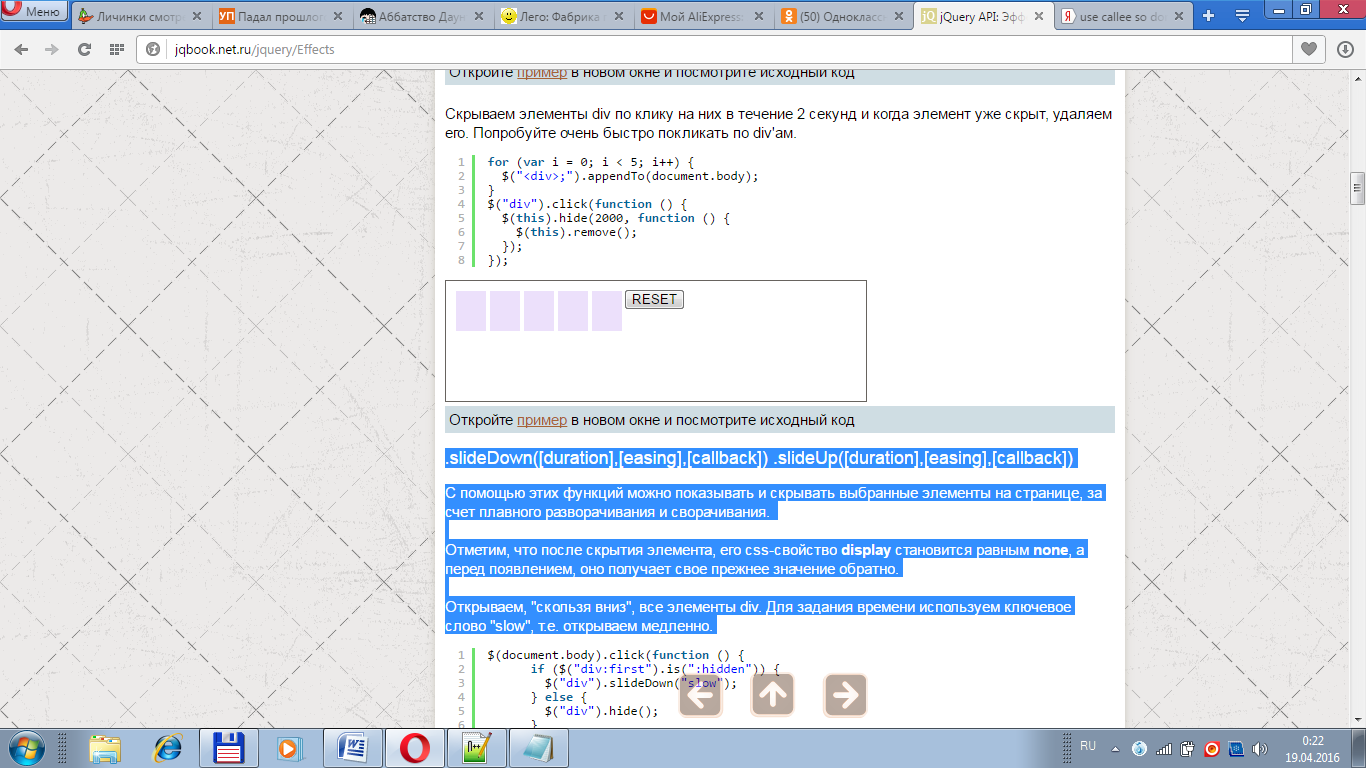


Пример 2: Показать все скрытые элементы div по порядку, используя ключевое слово fast. Каждая анимация занимает 200 миллисекунд и по завершении очередной стартует следующая.

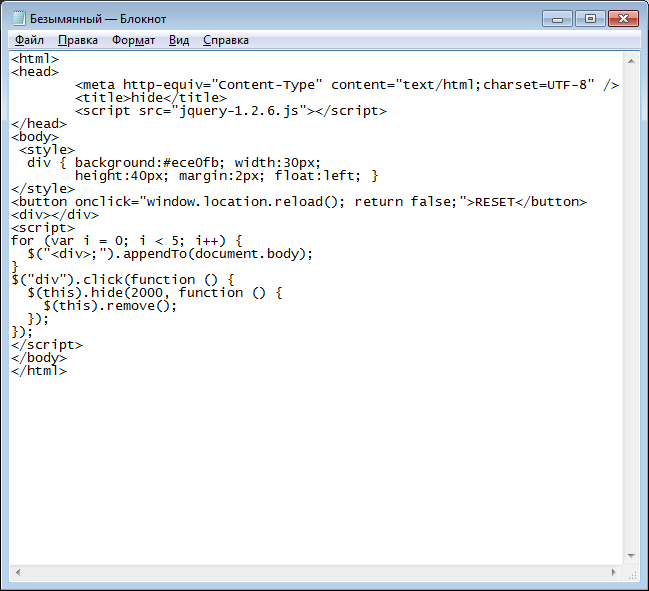
HTML-код:



Пример 3: Скрываем элементы div в течение 2 секунд по клику на них и когда элемент уже скрыт, удаляем его.



HTML-код:



**Методы свертывания**

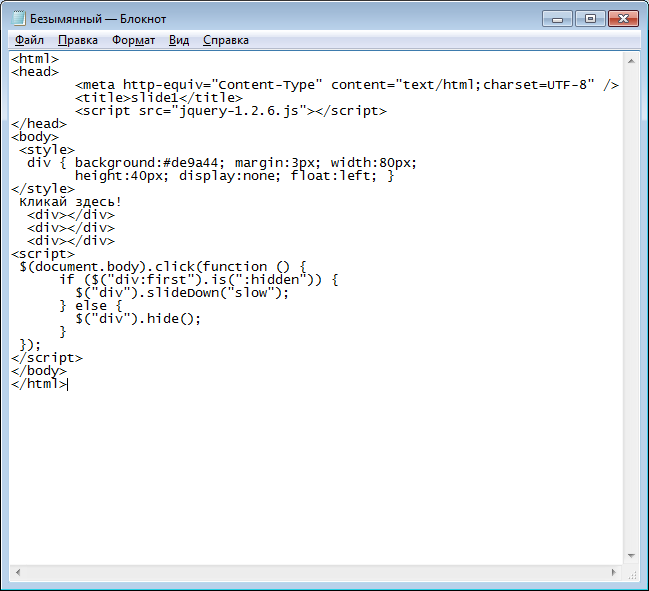
* **.slideDown([duration],[easing],[callback]) .slideUp([duration],[easing],[callback])**

С помощью этих функций можно показывать и скрывать выбранные элементы на странице, за счет плавного разворачивания и сворачивания.

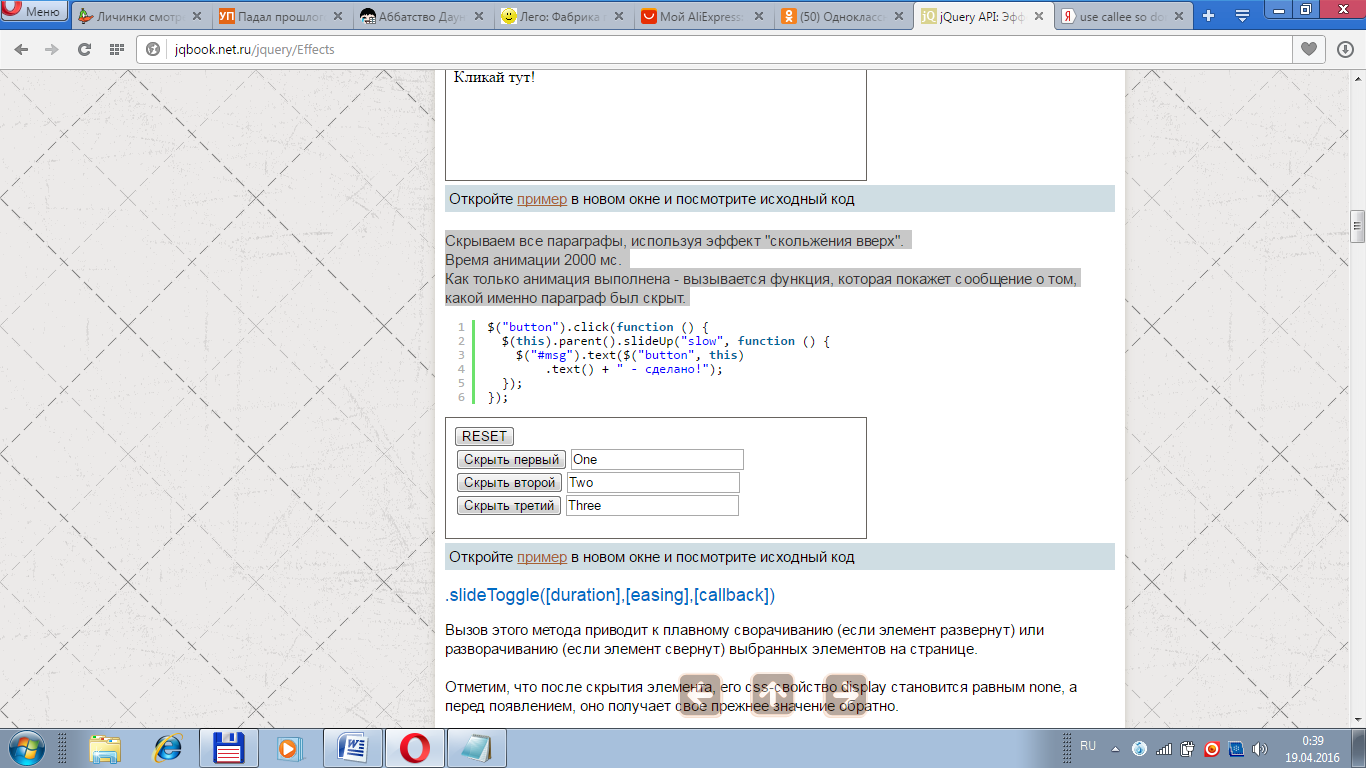
Отметим, что после скрытия элемента, его css-свойство **display** становится равным **none**, а перед появлением, оно получает свое прежнее значение обратно.

Пример 4: Открываем, "скользя вниз", все элементы div. Для задания времени используем ключевое слово "slow", т.е. открываем медленно.

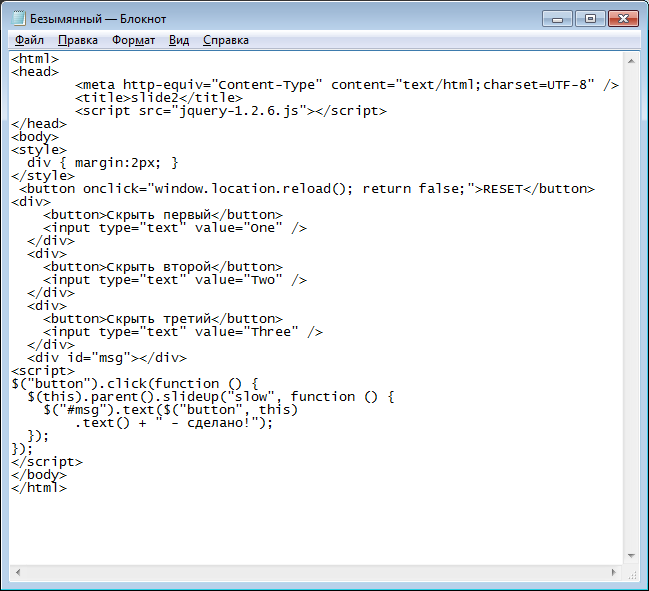
HTML-код:



Пример 5: Скрываем все параграфы, используя эффект "скольжения вверх". Время анимации 2000 мс.  Как только анимация выполнена - вызывается функция, которая покажет сообщение о том, какой именно параграф был скрыт.



HTML-код:



* **.slideToggle([duration],[easing],[callback])**

Вызов этого метода приводит к плавному сворачиванию (если элемент развернут) или разворачиванию (если элемент свернут) выбранных элементов на странице.

После скрытия элемента, его css-свойство display становится равным none, а перед появлением, оно получает свое прежнее значение обратно.

Примеры:

// свернет/развернет элемент с идентификатором leftFit за 400 мс.

$("#leftFit").slideToggle();

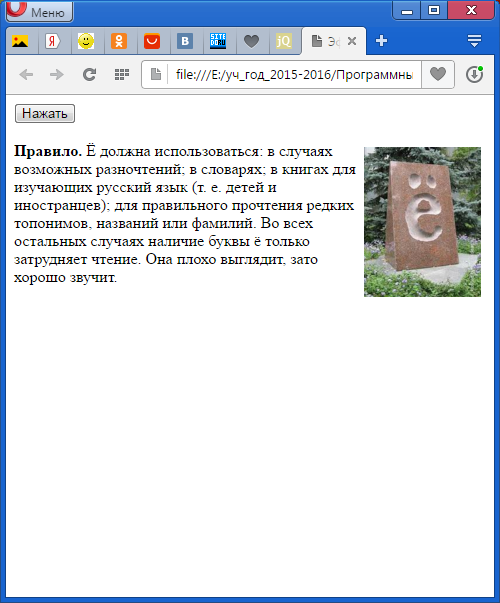
// в течении 1/3 секунды свернет/развернет элемент с идентификатором leftFit.

$("#leftFit").slideToggle(300);

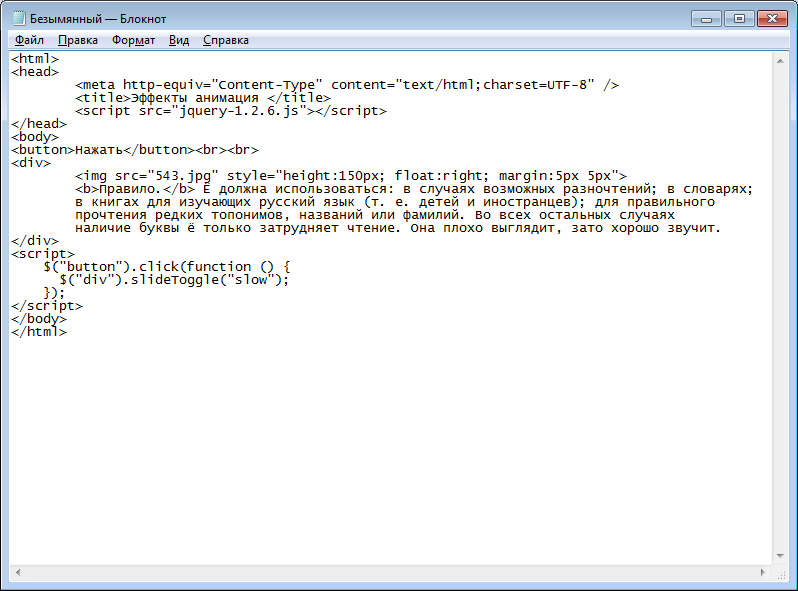
// в течении 600 мс свернет/развернет элемент с идентификатором leftFit.

$("#leftFit").slideToggle("slow");

Пример 6: При нажатии на кнопку элемент с картинкой и текстом будет поочередно сворачиваться и разворачиваться.



HTML-код:



**Методы исчезновения**

* **.fadeIn([duration],[easing],[callback])**
* **.fadeOut([duration],[easing],[callback])**

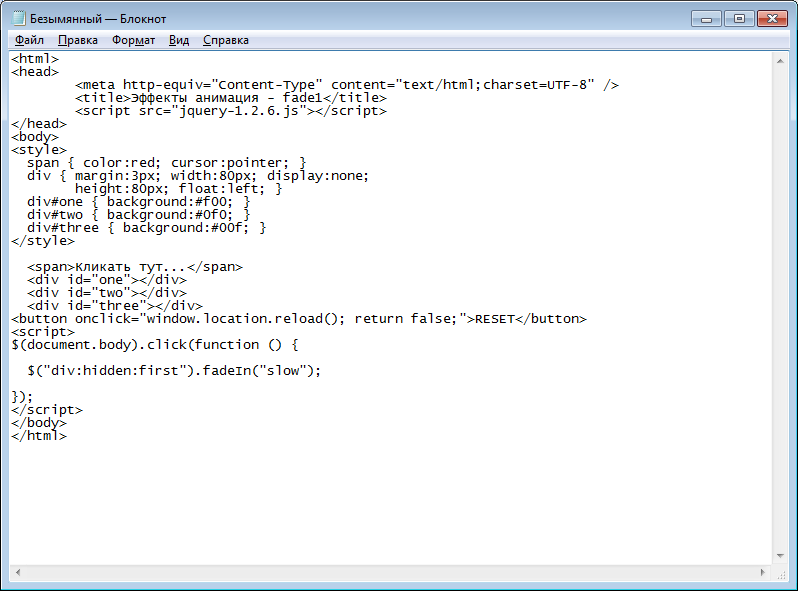
С помощью этих функций можно показывать и скрывать выбранные элементы на странице, за счет плавного изменения прозрачности.

Отметим, что после скрытия элемента, его css-свойство display автоматически становится равным none, а перед появлением, оно получает свое прежнее значение обратно.

Пример 7: "Проявляем" скрытые элементы div один за другим, каждый раз анимационный эффект занимает время примерно в 600 мс.

|  |  |
| --- | --- |
|  |  |

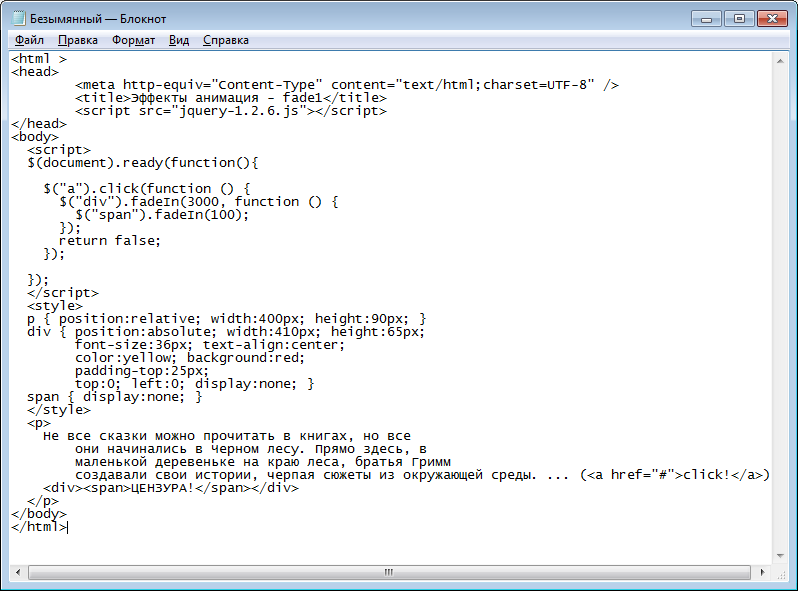
HTML-код:



Пример 8: "Проявляем" красный блок поверх текста. По завершении этой анимации запускается следующий эффект, который "проявляет" тэг span.

|  |  |
| --- | --- |
|  |  |

HTML-код:



* **.fadeTo([duration], opacity, [easing], [callback])**

Изменяет уровень прозрачности у выбранных элементов на странице. Позволяет менять прозрачность плавно.

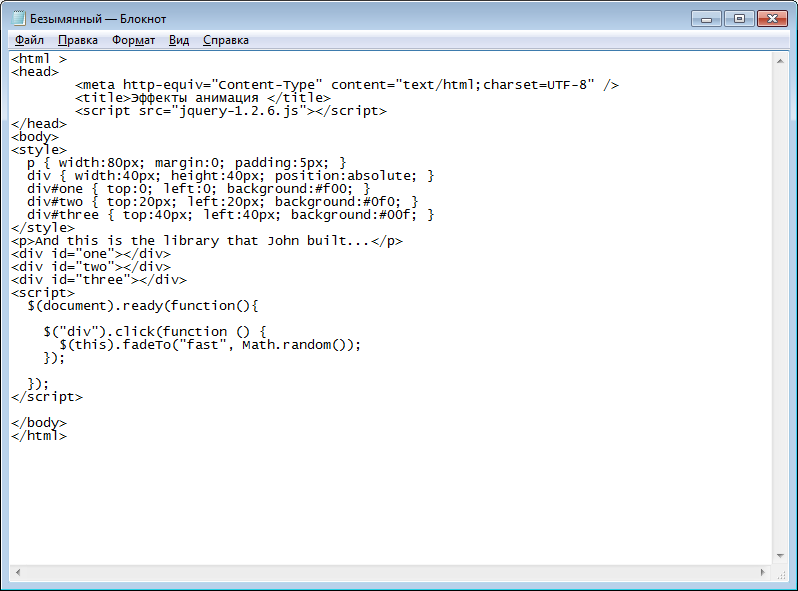
**fadeTo** принимает те же аргументы, что и show или hide.

И кроме того второй аргумент opacity - число от 0 до 1, характеризующий "целевую" прозрачность элемента.

Пример 9: Меняем прозрачность элементов div случайным образом.

|  |  |  |
| --- | --- | --- |
|  |  |  |

HTML-код:



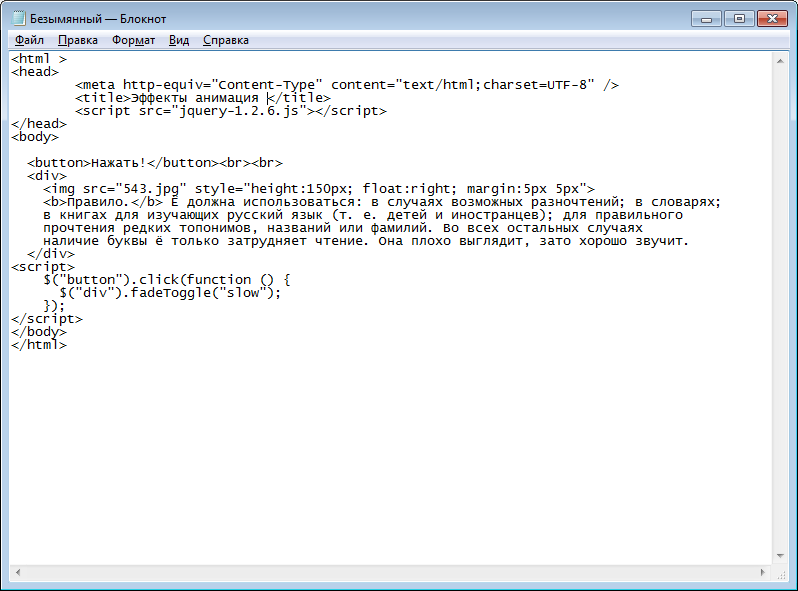
* **.fadeToggle([duration],[easing],[callback])**

Вызов этого метода приводит к плавному исчезновению (если элемент не скрыт) или появлению (если элемент скрыт) выбранных элементов на странице, за счет изменения прозрачности.

После скрытия элемента, его css-свойство display становится равным none, а перед появлением, оно получает свое прежнее значение обратно.

Пример 10: При нажатии на кнопку элемент с картинкой и текстом будет поочередно появляться и исчезать:

HTML-код:



**Методы анимации**

* **.animate(properties, [duration], [easing], [callback])**

Выполняет заданную пользователем анимацию, с выбранными элементами. Анимация происходит за счет плавного изменения CSS-свойств у элементов.

**properties**- список CSS-свойств, участвующих в анимации и их конечных значений.

Задается объектом, в формате css-свойство:значение

Это очень похоже на задание группы параметров в методе **.css().** Однако, **properties**имеет более **широкий диапазон** типов значений.

Они могут быть заданы не только в виде привычных единиц: чисел, пикселей, процентов и др., но еще и относительно:

{height:"+=30", left:"-=40"}

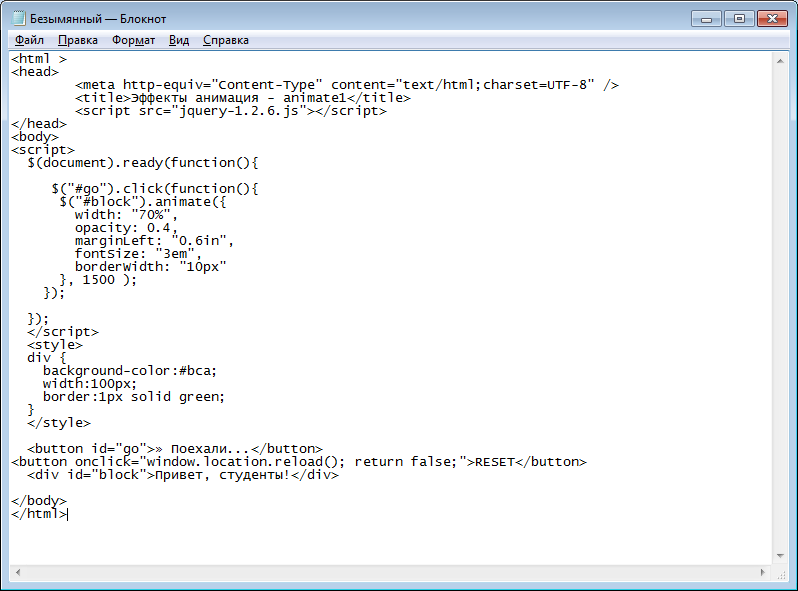
Кроме того, можно задавать значения "hide", "show", "toggle", которые скроют, покажут или изменят видимость элемента на противоположную, за счет параметра, к которому они применены.

Отметим, что в параметре properties можно указывать только те css-свойства, которые задаются с помощью числовых значений.  Например, свойство background-color использовать не следует.

Пример 11: Постепенно увеличим элемент div.

|  |  |
| --- | --- |
|  |  |

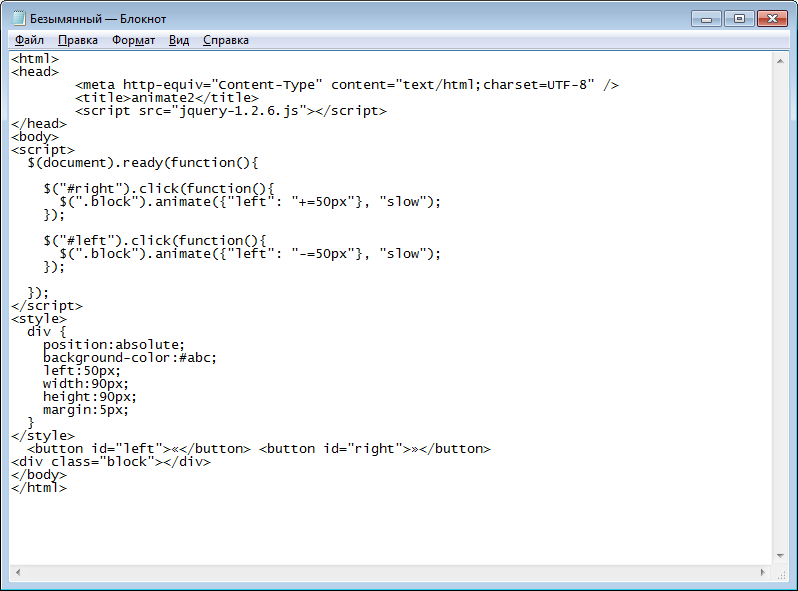
HTML-код:



Пример 12: Пример "относительной" анимации. Покликайте несколько раз по кнопкам вправо/влево.

|  |
| --- |
|  |

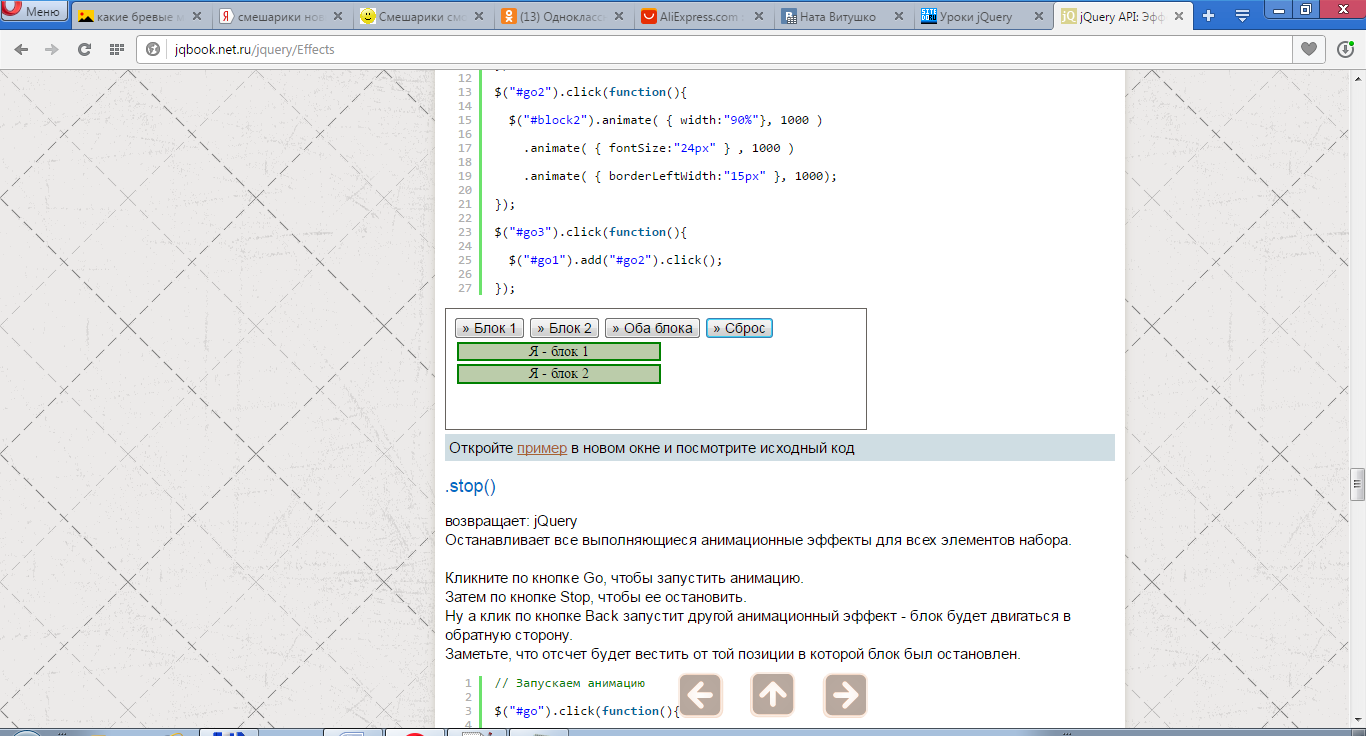
HTML-код:



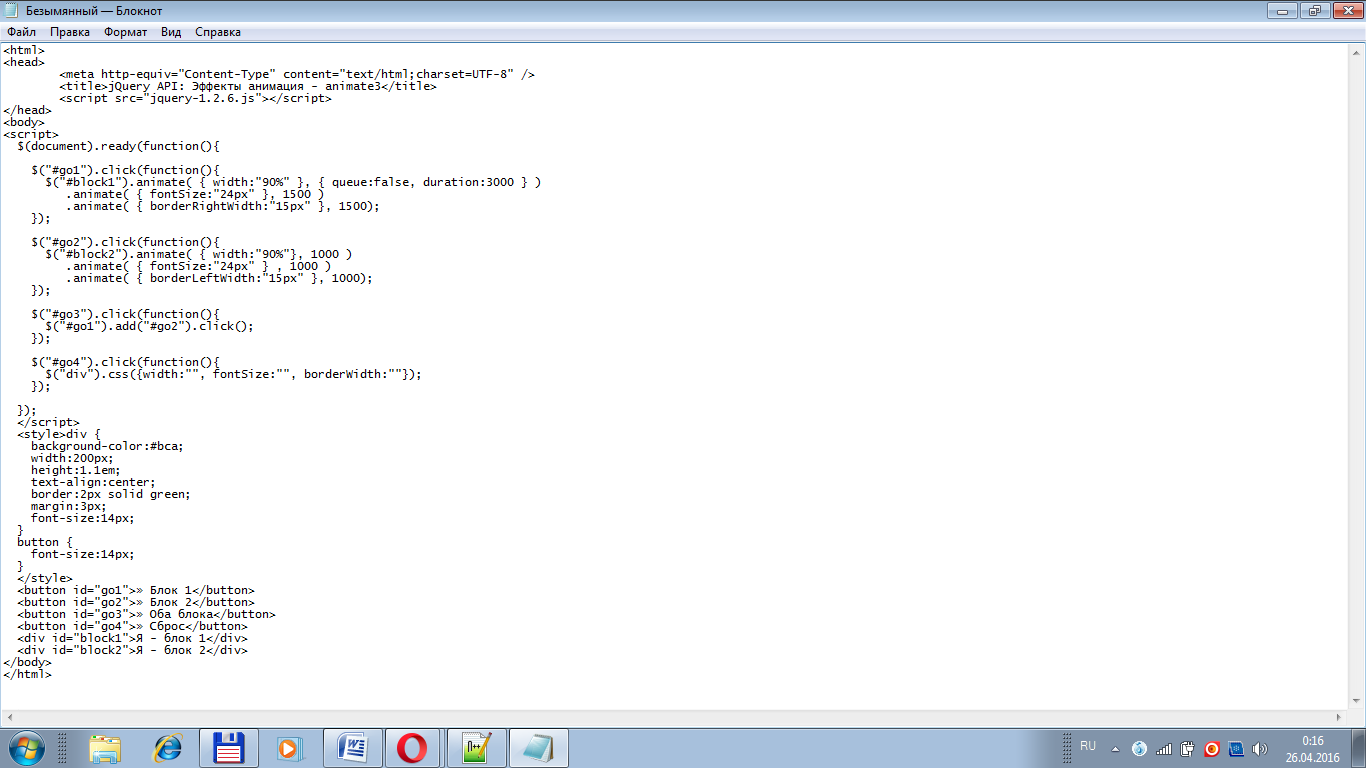
Пример 13: Некий элемент с идентификатором **#block1** должен изменить свою ширину до 90% от возможной, шрифт должен быть установлен в размер 24px и правая рамка должна стать шириной 15px.

Обратите внимание, что изменение ширины происходит одновременно с другими двумя эффектами, которые уже следуют друг за другом в порядке очереди.

Вторая кнопка запускает почти такую же анимацию, вот только все эффекты следуют друг за другом строго по очереди.



HTML-код:



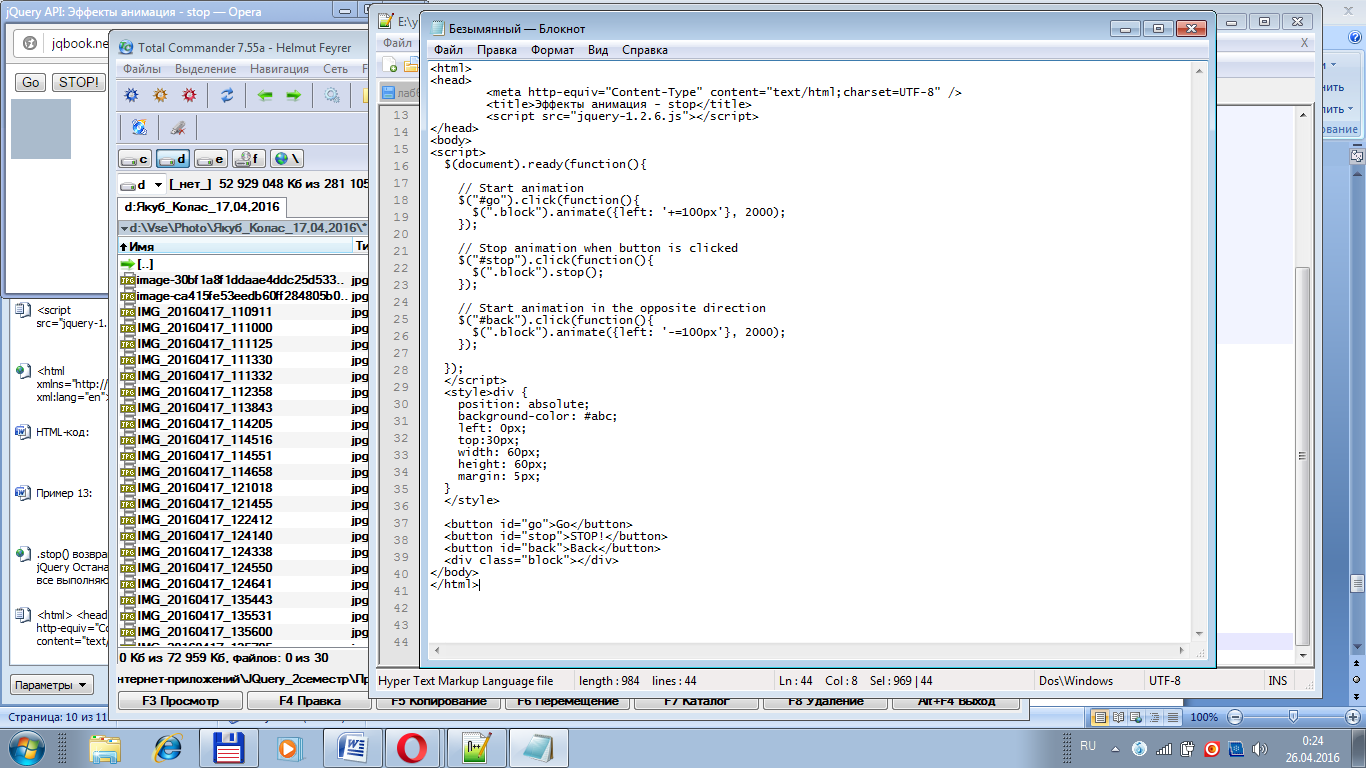
* **.stop()**

Останавливает все выполняющиеся анимационные эффекты для всех элементов набора.

Пример 14: Кликните по кнопке Go, чтобы запустить анимацию.  Затем по кнопке Stop, чтобы ее остановить.  Клик по кнопке Back запустит другой анимационный эффект - блок будет двигаться в обратную сторону. Отсчет будет вестить от той позиции, в которой блок был остановлен.

|  |
| --- |
|  |

HTML-код:



**Контрольные вопросы:**

1. Перечислите методы видимости. Каково их назначение?
2. Перечислите методы свертывания. Каково их назначение?
3. Перечислите методы исчезновения. Каково их назначение?
4. Перечислите методы анимации. Каково их назначение?

**Задания:**

1. Изучите теоретические сведения.
2. Изучите работу примеров на практике, исправьте ошибки в примерах.
3. Придумайте свой пример, демонстрирующие как можно больше анимационных эффектов.
4. Оформите отчет о лабораторной работе.

**Содержание отчета:**

Лабораторная работа № 6

Тема лабораторной работы.

Цель работы.

Краткие теоретические сведения (ответы на контрольные вопросы).

Вывод о проделанной работе.