jQuery для начинающих

* [jQuery](https://habr.com/ru/hub/jquery/)

[](http://anton.shevchuk.name/javascript/jquery-for-beginners/)  
  
jQuery — это замечательный JavaScript Framework, который подкупает своей простотой в понимании и удобством в использовании. Но изучение надо с чего-то начинать, и лично моё мнение — лучше всего начинать с наглядных примеров, и они далее…

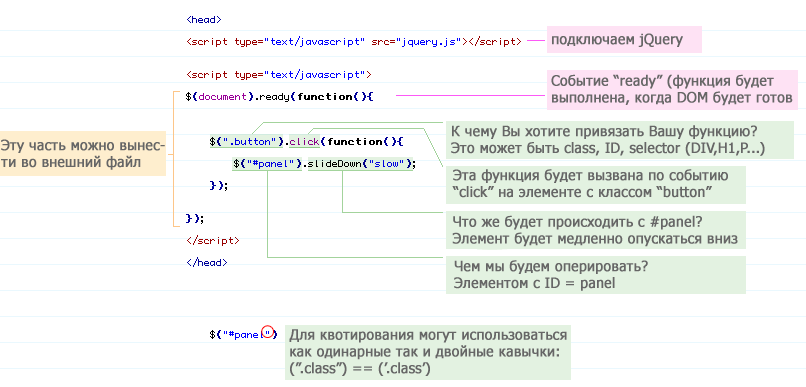
Как же все-таки работает jQuery?

Ну для начала Вам понадобиться сам фреймворк, его вы сможете скачать с [домашней страницы проекта](http://jquery.com/), затем проинициализировать:

<head>

<script type="text/javascript" src="jquery.js"></script>

</head>

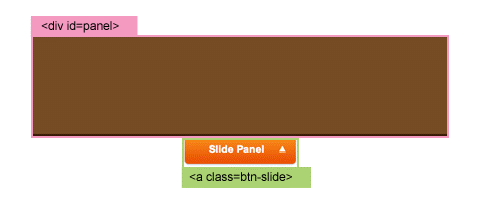
А основные моменты Вам поможет понять следующая диаграмма:  
 

Как получить элемент с помощью jQuery?

Для того чтобы понимать как работает селектор Вам все-же необходимы базовые знания CSS, т.к. именно от принципов CSS отталкивает селектор jQuery:

* $("#header") — получение элемента с id=«header»
* $(«h3») — получить все <h3> элементы
* $(«div#content .photo») — получить все элементы с классом =«photo» которые находятся в элементе div с id=«content»
* $(«ul li») — получить все <li> элементы из списка <ul>
* $(«ul li:first») — получить только первый элемент <li> из списка <ul>

Выдвижная панель

Начнем с простенького примера — слайд-панель, она у нас будет двигаться вверх/вниз по клику на ссылке (см. [пример](http://anton.shevchuk.name/wp-demo/jquery-tutorials/simple-slide-panel.html))  
  
Реализуем это следующим образом, по клику на ссылку, у нас будет переключаться её класс (между «active» и «btn-slide»), а панелька с id=«panel» будет выдвигаться/прятаться. (класс «active» изменяет позицию фонового изображения, см. CSS файл во вложении).

$(document).ready(function(){

$(".btn-slide").click(function(){

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

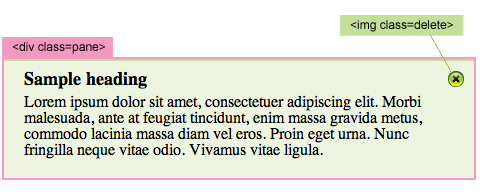
$(this).toggleClass("active");

});

});

Магические исчезновения

Этот пример покажет как можно красиво и легко ~~убирать~~ растворять элементы (см. [пример](http://anton.shevchuk.name/wp-demo/jquery-tutorials/simple-disappear.html)):  
  
  
Когда мы кликаем по картинке <img class=«delete»>, будет найден родительский элемент <div class=«pane»>, и его прозрачность будет медленно изменяться от opacity= 1.0 до opacity=hide:

$(document).ready(function(){

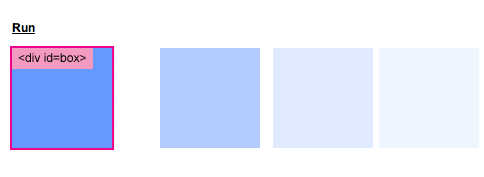
$(".pane .delete").click(function(){

$(this).parents(".pane").animate({ opacity: "hide" }, "slow");

});

});

Связанная анимация

Теперь пример посложнее, но он поможет Вам лучше понять jQuery. Всего несколько строк кода заставят квадрат двигаться, изменять размер и прозрачность. (см. [пример](http://anton.shevchuk.name/wp-demo/jquery-tutorials/chainable-effects.html)):  
  
Line 0: когда прогрузилась страница (DOM готов к манипуляциям)  
Line 1: привязываемся к событию click для элемента <a class=«run»>  
Line 2: манипулируем элементом <div id=«box»> — уменьшаем его прозрачность до 0.1, наращиваем позицию left еще на 400px, со скоростью 1200 (milliseconds)  
Line 3: затем медленно изменяем следующие параметры: opacity=0.4, top=160px, height=20, width=20; скорость анимации указывается вторым параметром: «slow», «normal», «fast» или в миллисекундах  
Line 4: затем opacity=1, left=0, height=100, width=100, скорость — «slow»  
Line 5: затем opacity=1, left=0, height=100, width=100, скорость — «slow»  
Line 6: затем top=0, скорость — «fast»  
Line 7: затем slideUp (с дефолтной скоростью анимации — «normal»)  
Line 8: затем slideDown, скорость — «slow»  
Line 9: возвращаем false для того чтобы браузер не перешел по ссылке

$(document).ready(function(){

$(".run").click(function(){

$("#box").animate({opacity: "0.1", left: "+=400"}, 1200)

.animate({opacity: "0.4", top: "+=160", height: "20", width: "20"}, "slow")

.animate({opacity: "1", left: "0", height: "100", width: "100"}, "slow")

.animate({top: "0"}, "fast")

.slideUp()

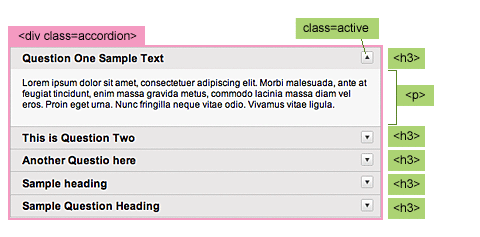
.slideDown("slow")

return false;

});

});

Гармошка #1

Пример реализации «гармошки». (см. [пример](http://anton.shevchuk.name/wp-demo/jquery-tutorials/accordion1.html))  
  
Теперь приступим к разбору полетов:  
Первой строчкой мы добавляем класс «active» первому элементу <h3> внутри <div class=«accordion»> (класс«active» отвечает за позиционирования фонового рисунка — иконки со стрелочкой). Во второй строчке мы прячем все не первые <p> элементы внутри <div class=«accordion»>.  
Когда происходит клик по заголовку <h3>, для следующего в нём элемента <p> будет применен эффект slideToggle, затем для всех остальных элементов <p> будет применен эффект slideUp. Следующие действие изменяет класс заголовка на «active», затем ищем все остальные заголовки <h3> и убираем у них класс «active»

$(document).ready(function(){

$(".accordion h3:first").addClass("active");

$(".accordion p:not(:first)").hide();

$(".accordion h3").click(function(){

$(this).next("p").slideToggle("slow")

.siblings("p:visible").slideUp("slow");

$(this).toggleClass("active");

$(this).siblings("h3").removeClass("active");

  });

 });

Гармошка #2

Этот пример схож с предыдущим, лишь отличается тем, что мы указываем открытую по умолчанию панельку. (см. [пример](http://anton.shevchuk.name/wp-demo/jquery-tutorials/accordion2.html))  
  
В CSS у нас указано для всех элементов <p> display:none. Теперь нам необходимо открыть третью панель. Для этого мы можем написать следующее $(".accordion2 p").eq(2).show(), где eq обозначает равенство. Помните, что индексирование начинается с нуля:

$(document).ready(function(){

$(".accordion2 h3").eq(2).addClass("active");

$(".accordion2 p").eq(2).show();

$(".accordion2 h3").click(function(){

$(this).next("p").slideToggle("slow")

.siblings("p:visible").slideUp("slow");

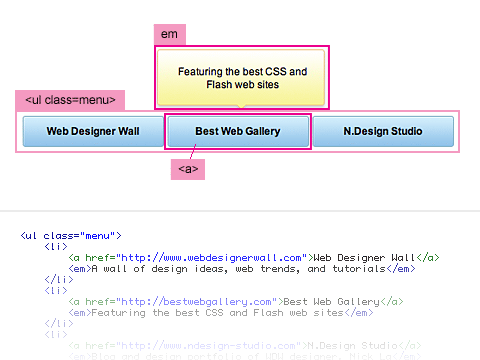
$(this).toggleClass("active");

$(this).siblings("h3").removeClass("active");

});

});

Анимация для события hover #1

Данный пример поможет создать Вам очень красивую анимацию для события hover (надеюсь, [Вы знаете что это](http://www.w3.org/TR/REC-CSS2/selector.html)?), (см. [пример](http://anton.shevchuk.name/wp-demo/jquery-tutorials/animated-hover1.html)):  
  
  
Когда Вы наводите мышкой на элемент меню (mouseover), происходит поиск следующего элемента <em>, и анимируется его прозрачность и расположение:

$(document).ready(function(){

$(".menu a").hover(function() {

$(this).next("em").animate({opacity: "show", top: "-75"}, "slow");

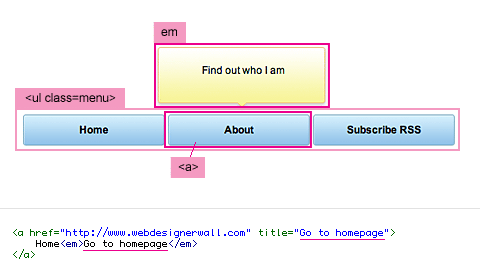
}, function() {

$(this).next("em").animate({opacity: "hide", top: "-85"}, "fast");

});

});

Анимация для события hover #2

Данный пример чуть-чуть посложней предыдущего примера: для формирования подсказки используется атрибут linktitle (см. [пример](http://anton.shevchuk.name/wp-demo/jquery-tutorials/animated-hover2.html))  
  
Первым делом добавим тэг <em> в каждый элемент <a>. Когда произойдет событие mouseover, мы возьмем текст из атрибута «thetitle» и сохраним его в переменной «hoverText», затем этот текст будет вставлен в тэг <em>:

$(document).ready(function(){

$(".menu2 a").append("<em></em>");

$(".menu2 a").hover(function() {

$(this).find("em").animate({opacity: "show", top: "-75"}, "slow");

var hoverText = $(this).attr("title");

$(this).find("em").text(hoverText);

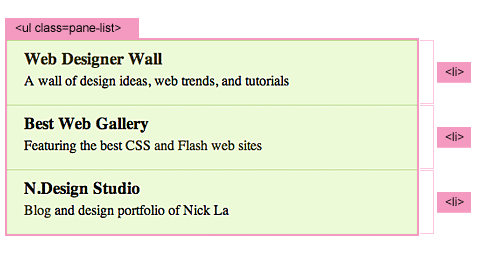
}, function() {

$(this).find("em").animate({opacity: "hide", top: "-85"}, "fast");

});

});

Кликабельные блоки

Этот пример демонстрирует как сделать кликабельным блок с текстом, а не только ссылку (см. [пример](http://anton.shevchuk.name/wp-demo/jquery-tutorials/block-clickable.html)):  
  
Создадим список <ul> с классом class=«pane-list» и мы хотим сделать элементы <li> кликабельными. Для начала привяжемся к событию click для элемента ".pane-list li"; когда пользователь будет кликать по элементу списка, наша функция произведет поиск тэга <a> и сделает редирект на страницу указанную в атрибуте href.

$(document).ready(function(){

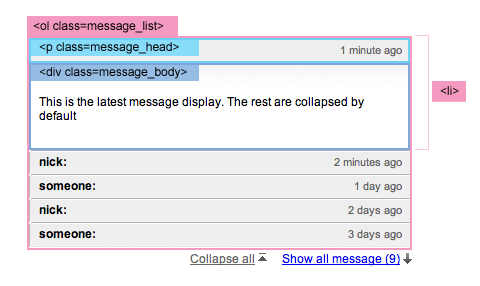
$(".pane-list li").click(function(){

window.location=$(this).find("a").attr("href"); return false;

});

});

Складывающиеся панельки

Ну а теперь чуть-чуть скомбинируем предыдущие примеры и создадим ряд складывающихся панелек (наподобие как в Gmail организован inbox). (см. [пример](http://anton.shevchuk.name/wp-demo/jquery-tutorials/collapsible-panels.html))  


* скрываем все элементы <div class=«message\_body»> после первого.
* скрываем все элементы <li> после пятого
* клик по <p class=«message\_head»> — вызывает метод slideToggle для следующего элемента <div class=«message\_body»>
* клик по <a class=«collpase\_all\_message»> — вызывает метод slideUp для всех <div class=«message\_body»>
* клик по <a class=«show\_all\_message»> — скрывает элемент, и отображает <a class=«show\_recent\_only»>, так же вызывается метод slideDown для всех <li> послепятого
* клик по <a class=«show\_recent\_only»> — скрывает элемент, и отображает <a class=«show\_all\_message»>, так же вызывается метод slideUp для всех <li> после пятого

$(document).ready(function(){

//hide message\_body after the first one

$(".message\_list .message\_body:gt(0)").hide();

//hide message li after the 5th

$(".message\_list li:gt(4)").hide();

//toggle message\_body

$(".message\_head").click(function(){

$(this).next(".message\_body").slideToggle(500)

return false;

});

//collapse all messages

$(".collpase\_all\_message").click(function(){

$(".message\_body").slideUp(500)

return false;

});

//show all messages

$(".show\_all\_message").click(function(){

$(this).hide()

$(".show\_recent\_only").show()

$(".message\_list li:gt(4)").slideDown()

return false;

});

//show recent messages only

$(".show\_recent\_only").click(function(){

$(this).hide()

$(".show\_all\_message").show()

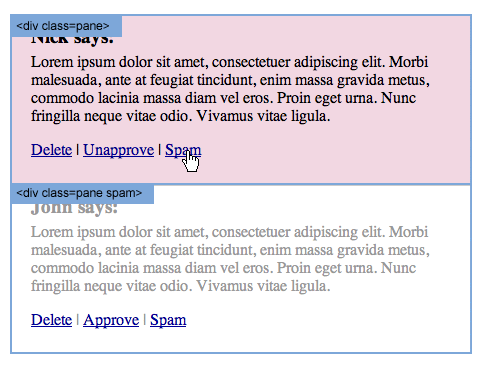
$(".message\_list li:gt(4)").slideUp()

return false;

});

});

Имитация Backend'a Wordpress'a

Я думаю многие из читателей сталкивались с админской частью wordpress'a, точнее с редактирование комментариев. Попробуем сделать что-то подобное. Для анимации фонового цвета нам понадобиться соответствующий [плагин для jQuery](http://plugins.jquery.com/project/color). (см. [пример](http://anton.shevchuk.name/wp-demo/jquery-tutorials/wordpress-comments.html))  


* добавим класс «alt» к каждому чётному элементу <div class=«pane»> (данный класс изменяет цвет фона элемента)
* клик по <a class=«btn-delete»> инициирует появление сообщения (alert), так же происходит анимация фонового цвета и прозрачности (backgroundColor и opacity) для <div class=«pane»>
* клик по <a class=«btn-unapprove»> — вызывает анимацию фона у <div class=«pane»> (цвет изменяется на желтый и обратно) и добавляет класс «spam»
* клик по <a class=«btn-approve»> — вызывает анимацию фона у <div class=«pane»> (цвет изменяется на зеленый и обратно) и удаляет класс «spam»
* клик по <a class=«btn-spam»> — вызывает анимацию фона у <div class=«pane»> (цвет изменяется на красный), и изменяет параметр «opacity» на «hide»

//don't forget to include the Color Animations plugin

$(document).ready(function(){

$(".pane:even").addClass("alt");

$(".pane .btn-delete").click(function(){

alert("This comment will be deleted!");

$(this).parents(".pane").animate({ backgroundColor: "#fbc7c7" }, "fast")

.animate({ opacity: "hide" }, "slow")

return false;

});

$(".pane .btn-unapprove").click(function(){

$(this).parents(".pane").animate({ backgroundColor: "#fff568" }, "fast")

.animate({ backgroundColor: "#ffffff" }, "slow")

.addClass("spam")

return false;

});

$(".pane .btn-approve").click(function(){

$(this).parents(".pane").animate({ backgroundColor: "#dafda5" }, "fast")

.animate({ backgroundColor: "#ffffff" }, "slow")

.removeClass("spam")

return false;

});

$(".pane .btn-spam").click(function(){

$(this).parents(".pane").animate({ backgroundColor: "#fbc7c7" }, "fast")

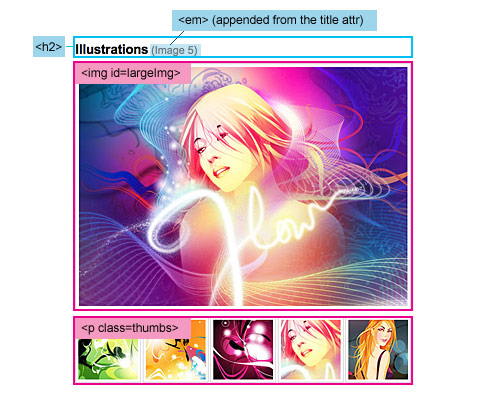
.animate({ opacity: "hide" }, "slow")

return false;

});

});

Галерея изображений

Простейший пример реализации галереи, без перезагрузки страницы. (см. [пример](http://anton.shevchuk.name/wp-demo/jquery-tutorials/img-replacement.html))  
  
 Для начала добавим тэг <em> в заголовки <h2>  
 По клику на изображения в <p class=thumbs> выполняем следующие действия:

* сохраняем значение атрибута «href» в переменной «largePath»
* сохраняем значение атрибута «title» в переменной «largeAlt»
* заменяем в элементе <img id=«largeImg»> значение атрибута «scr» и «alt» значениями из переменных «largePath» и «largeAlt»
* так же присваиваем элементу «h2 em» значение из «largeAlt»

$(document).ready(function(){

$("h2").append('<em></em>')

$(".thumbs a").click(function(){

var largePath = $(this).attr("href");

var largeAlt = $(this).attr("title");

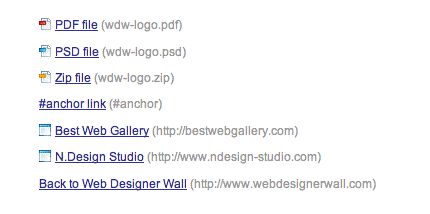
$("#largeImg").attr({ src: largePath, alt: largeAlt });

$("h2 em").html(" (" + largeAlt + ")"); return false;

});

 });

Стилизируем ссылки

Большинство нормальных браузеров легко понимают когда мы хотим добиться от них стилизации ссылок для различного типа файлов, для это цели можно использовать следующее CSS правило: a[href $='.pdf'] {… }. Но как обычно IE6 отличается умом и сообразительностью, по этой причине будем ставить ему костыли используя jQuery. (см. [пример](http://anton.shevchuk.name/wp-demo/jquery-tutorials/link-types.html))   
  
Для начала добавим класс для каждой ссылки, в соответствии с типом файла.   
Затем выберем все элементы <a> которые не содержат ссылки на "[http://www.webdesignerwall.com](http://www.webdesignerwall.com/)" и не начинающиеся на "#" в «href», затем добавим им класс «external» и устанавливаем target= "\_blank".

$(document).ready(function(){

$("a[@href$=pdf]").addClass("pdf");

$("a[@href$=zip]").addClass("zip");

$("a[@href$=psd]").addClass("psd");

$("a:not([@href\*=http://www.webdesignerwall.com])").not("[href^=#]")

.addClass("external")

.attr({ target: "\_blank" });

});