**Краткое пособие по основам jQuery.**

В данном пособии на примерах будет очень кратко изложены возможности **jQuery**, данного объема знаний хватит на то, чтоб после самостоятельно без проблем и трудностей при помощи стандартной документации любой мог сделать с **jQuery** то, что от нее потребуется.

Пособие написано специально для **snakeproject.ru.**

Критику и отзывы можно оставить в форме на страничке «контакты» на сайте.

Желаю приятного обучения.

Для начала создадим два файла:

**Index.php – страница сайта:**

<html>

<head>

<title></title>

<link rel="stylesheet" href="../css/style.css" type="text/css" />

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

<script type="text/javascript" src="5graph.js" ></script>

</head>

<body>

<div id='idDiv'>

<ul>

<li>One</li>

<li class='newClass2'>Two</li>

<li>Three</li>

<li class='newClass2'>Four</li>

<li>Five</li>

<li class='newClass2'>Six</li>

</ul>

</div>

<hr />

<style>

.newClass{

background: yellow;

}

.newClass2{

font-family: Arial;

font-weight: bold;

}.newClass3{

display: none;

type: hidden;

}

</style>

</body>

</html>

**Style.css – страница стилей:**

body {

font-family: arial;

color: maroon;

background: silver;

font-size: 14px;

}

Плюс необходимо скачать саму библиотеку **jQuery** по адресу: <http://jquery.com/>

И к последнему примеру необходимо скачать **плагин**: <http://graph.prootime.ru/>

**Для начала у нас все готово, приступим!**

1. **Как и когда будем выполнять код?**

С помощью данной конструкции мы будем ожидать полной загрузки документа, и только после мы будем проделывать с ней какой-либо код. Сам код будет записываться между фигурными скобками {}:

//После загрузки документа исполняется функция

<script>

$(function(){

});

</script>

1. **Обращение к HTML тегам**

**Напишем между { и } следующий код. Тут собраны некоторые примеры, как нам обратиться к HTML тегу, по его имени\id\дочерним тегам\первым\последним и т.п. В конце мы выводим в консоль то, что получили из выборки(в google chrome нажать f12 и пункт console):**

**//Обращение к тегам**

$('li');

$('#idDiv');

$('ul li');

$('li:first');

$('li:last');

$('li:even'); //Нечетные

$('li:odd'); //Четные

**//Обращение к div**

var x = $('div');

// Все li внутри div

var y = $('li', x);

**//Если выборка не пустая**

if(x.length>0)

console.log(x);

1. **Отлавливаем по какому элементу на странице произошел клик**

$(document). – указываем, что работаем с телом страницы.

on('click' – связываем с событием клика мышкой по телу документа

function(per){console.log(per.target); - вызывается функция, которой передается в аргумент то, по чему был клик мышкой, и выводим результат в консоль.

**//Отлавливаем по какому элементу произошел клик**

$(document).on('click', function(per){

console.log(per.target);

});

1. **Работаем с CSS**

**Далее мы увидим примеры:**

**//Обращение к CSS свойствам, их установка и замена:**

$('ul li').css('font-size', '16px');

$('ul li').css({

'color': 'DarkGreen',

'background': 'white'

});

**//Добавление класса:**

$('div').addClass('newClass');

//Удаление класса

$('div').removeClass('newClass');

//Удаляет, если есть класс, и добавляет, если класса нет

$('ul li').toggleClass('newClass2');

**//Если теги видимые, то удаляем у четных класс и добавляем им другой класс:**

if($('ul li').is(':visible')){

$('li:odd').removeClass('newClass2');

$('li:odd').addClass('newClass3');

}

**//Ситуация с точностью наоборот:**

if($('ul li:odd').is(':hidden')){

$('li:odd').removeClass('newClass3');

$('li:odd').addClass('newClass2');

}

**//Ситуация с "отрицательным" условием:**

if($('ul li:odd').not(':hidden')){

$('li:odd').removeClass('newClass2');

$('li:odd').addClass('newClass3');

}

1. **Срытие и вывод блока на экран:**

**Перепишем index.php:**

<div id='idDiv'>

<p><input type='submit' id='buttonHide' name='buttonHide' value='Скрыть блок' ></p>

</div>

<div id='hDiv'>

<p>Некоторый блок</p>

<p>Который нужно скрыть</p>

</div>

**В тегах “script” перепишем код:**

//Реакция на событие **click** у кнопки **buttonHide** для сокрытия или вывода блока.

С помощью **is** мы проверяем условие на видимость блока **div,** а с помощью **val** –меняем надпись на кнопке.

$('#buttonHide').on('click', function(){

if($('#hDiv').is(':visible')){

$('#hDiv').hide();

$(this).val('Показать блок');

}

else{

$('#hDiv').show();

$(this).val('Скрыть блок');

}

});

**//Плавное сокрытие или вывод блока**

$('#buttonHide').on('click', function(){

if($('#hDiv').is(':visible')){

$('#hDiv').slideUp();

$(this).val('Показать блок');

}

else{

$('#hDiv').slideDown();

$(this).val('Скрыть блок');

}

});

**//Еще плавное сокрытие или вывод блока**

$('#buttonHide').on('click', function(){

if($('#hDiv').is(':visible')){

$('#hDiv').fadeOut();

$(this).val('Показать блок');

}

else{

$('#hDiv').fadeIn();

$(this).val('Скрыть блок');

}

});

**//Еще плавное сокрытие или вывод блока**

$('#buttonHide').on('click', function(){

$('#hDiv').slideToggle(2000);

});

**//Еще плавное сокрытие или вывод блока**

$('#buttonHide').on('click', function(){

$('#hDiv').fadeToggle(2000);

});

**//Еще плавное сокрытие или вывод блока**

$('#buttonHide').on('click', function(){

$('#hDiv').fadeToggle('slow');

});

**//Еще плавное сокрытие или вывод блока**

$('#buttonHide').on('click', function(){

$('#hDiv').fadeToggle('fast');

});

1. **Обращение к «детям» тега и изменение его содержимого**

**//Обращение к внутреннему содержимому четвертого li внутри ul**

console.log($('ul').children('li').eq(3).text());

**//Изменение внутреннего содержимого**

'ul').children('li').eq(3).text('newFour');

**//Изменение внутреннего содержимого пердыдущего элемента**

'ul').children('li').eq(4).prev().text('newNewFour');

1. **Изменение значений формы**

**Перепишем index.php:**

<div>

<form name='form' action='index.php' method='POST'>

<p>Введите данные: <input type='text' name='t1' size='5'></p>

<p>Введите данные: <input type='text' name='t2' size='5' value='Text2'></p>

<p><input type='submit' name='button' ></p>

</form>

</div>

**Перепишем код javascript:**

**С помощью конструкции** 'input[type="text**"] мы обращаемся к тегу** <input> **с типом** type="text":

**//Добавим значения во все элементы формы**

$('input[type="text"]').val('Text');

**//Добавим значение в поле по имени**

$('input[name="t2"]').val('newText2');

**//Получим значение из поля по имени**

console.log($('input[name="t2"]').val());

1. **Анимация**

**Разберем простейший пример анимации:**

**Перепишем index.php:**

<div>

<p><b>Анимация.</b></p>

<p id="p">Движущийся абзац.</p>

<input type="button" id="b" value="Начать анимацию" />

</div>

**Перепишем код JavaScript:**

//Анимация. По действиям разложены изменения тегов, 2000 – параметр «периода времени»

$("#b").click(function(){

$("#p").animate({fontSize:"2.2em"},2000);

$("#p").animate({marginLeft:"30px"},2000);

$("#p").animate({marginLeft:"0px"},2000);

$("#p").animate({fontSize:"1em"},2000);

});

1. **Графики**

**Тут мы рассмотрим пример построения графиков с помощью плагина 5graph.**

**В начале пособия даны ссылки на скачивание, там и полная документация, плагин прост в освоении.**

**Перепишем index.php:**

<div>

<p><b>Графики:</b><a href="http://graph.prootime.ru/generator"> 5graph</p>

<script>

**//Сейчас я дам пример кода с двумя графиками, но на сайте проекта Вы можете сделать //любой другой вариант в зависимости от Ваших нужд.**

</script>

**//Определяем canvas**

<canvas id="canv" width="980" height="300"></canvas>

</div>

**//Вот мой пример кода с 5graph**

var properties = {

"background":"lightGray", //общий фон

"linescolor":"#ffffff", //цвет сетки

"lineswidth":3, //ширина линий сетки

"gridwidth":201, //минимальное расстояние между вертикальными линиями

"gorizontfontcolor":"blue", //цвет текста на горизонтальных линиях

"verticalfontcolor":"darkGreen", //цвет текста на вертикальных линиях

"gorizontfont":"bold 12px Arial", //шрифт на горизонтальных линиях

"verticalfont":"bold 12px Arial", //шрифт на вертикальных линиях

};

var hints = {

'background':'#4a4a4a', //фон подсказки

'bordercolor':'#000', //цвет границы

'borderwidth':2, //ширина границы

'paddingtb':2, //отступы сверху и снизу

'paddinglr':5, //отступы слева и справа

'borderradius':6, //закругленность прямоугольника

'lineheight':2, //расстояние между строками

'font':[ //шрифты построчно

{'color':'#fff','font':'bold 12px Verdana'},

{'color':'#fff','font':'bold 12px Verdana'}

]

};

//массив для построения графика

var data = {'graph': [

{'point': [

{"val": 40, "gval":"10 июня", "desc": "Строка 1"},

{"val": 60, "gval":"11 июня", "desc": "Строка 1"},

{"val": 50, "gval":"12 июня", "desc": "Строка 1"},

{"val": 10, "gval":"13 июня", "desc": "Строка 1"}

],

'properties':{

'linecolor':'#8cb9c8', //цвет линий соединящих точки графика

'pointcolor':'#8cb9c8', //цвет точек

'spacecolor':'#8cb9c8', //цвет области под графиком

'linewidth':2, //толщина линий соединяющие точки

'pointradius':3, //радиус точек

'spacealpha':0.1, //степень прозрачности области под графиком

'activepointwidth':4 //радиус активной точки

}

},

{'point': [

{"val": 20, "gval":"10 июня", "desc": "Строка 2"},

{"val": 40, "gval":"11 июня", "desc": "Строка 2"},

{"val": 70, "gval":"12 июня", "desc": "Строка 2"},

{"val": 40, "gval":"13 июня", "desc": "Строка 2"},

{"val": 30, "gval":"14 июня", "desc": "Строка 2"}

],

'properties':{

'linecolor':'#F1AA7A', //цвет линий соединящих точки графика

'pointcolor':'#F1AA7A', //цвет точек

'spacecolor':'#F1AA7A', //цвет области под графиком

'linewidth':2, //толщина линий соединяющие точки

'pointradius':3, //радиус точек

'spacealpha':0.1, //степень прозрачности области под графиком

'activepointwidth':4 //радиус активной точки

}

}]};

//И в конце вызываем функцию **plotgraph**

plotgraph('#canv',data,properties,hints)

**На этом все, исходники примеров я заархивирую и выложу на сайте.**