Лабораторная работа 2\_4. Анимация в JQuery

**Анимация в jQuery**

С помощью метода **animate()** можно создавать полноценную анимацию.

Синтаксис:

$("селектор").animate({стили},скорость,функция\_смягчения,функция),

где:

стили – CSS стили для анимации (к элементу одновременно может быть применено несколько стилей);

скорость – скорость анимации. Можно указать скорость с помощью предопределенных свойств: "slow", "fast", "normal" (медленно, быстро, нормально) или указать скорость в миллисекундах;

функция\_смягчения – функция, которая будет отвечать за плавность выполнения анимации;

функция – имя функции, код которой будет выполнен после завершения анимации.

Пример:

$(document).ready(function(){

$("#but1").click(function(){

$("p").animate({fontSize:30},2000);

$("p").animate({top:220},2000);

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

$("p").animate({left:320},2000);

$("p").animate({top:0,left:0},2000);

});

});

Следующий пример демонстрирует простое использование метода animate(), выполняется перемещение элемента <div> вправо, пока левое свойство элемента 250px.

<!DOCTYPE html>

<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>

<script>

$(document).ready(function(){

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

$("div").animate({left: '250px'});

});

});

</script>

</head>

<body>

<button>Start Animation</button>

<p>By default, all HTML elements have a static position, and cannot be moved. To manipulate the position, remember to first set the CSS position property of the element to relative, fixed, or absolute!</p>

<div style="background:#98bf21;height:100px;width:100px;position:absolute;"></div>

</body>

</html>

**Управление стилями в jQuery**. Метод **css()**.

jQuery имеет группу различных методов значительно упрощающих оформление элементов. Одним из самых важных методов в этой группе является метод **css()**.

С помощью метода **css** можно узнавать текущие или устанавливать новые значения свойств оформления элементов.

Синтаксис:

//Узнаем значение указанного CSS свойства выбранного элемента

$("селектор").css("свойство");

//Установим новое значение указанному CSS свойству элемента

$("селектор").css("свойство","значение");

//Установим произвольные значения нескольким CSS свойствам выбранного элемента

$("селектор").css({свойство1:значение1, свойствоN: значениеN});

Например,

$(document).ready(function(){

$("#but1").click(function(){

alert($("#par1").css("color"));

});

$("#but2").click(function(){

$("#par2").css("color","red");

});

$("#but3").click(function(){

$("#par3").css({"font-size":"27px","color":"red","font-family":"Arial"});

});

$("#text1").focus(function(){

$("#text1").val("");

});

$("#but4").click(function(){

$("#par4").css("font-size",$("#text1").val()+"px");

});

});

Обратите внимание, что для метода animate() можно определить несколько свойств для анимирования:

$("button").click(function(){ $("div").animate({

left: '250px',

opacity: '0.5',

height: '150px',

width: '150px' });

});

Использование относительных значений

Также можно определить относительные значения (тогда значение будет относиться к текущему значению элемента). Это делается путем помещения + = или - = перед значением:

пример

$("button").click(function(){ $("div").animate({

left: '250px',

height: '+=150px',

width: '+=150px' }); });

Задание.

1. реализовать открытку с анимационными эффектами - падающий дождь, розы, и пр.

2. на странице находится 4 квадрата. При нажатии на поле квадрата, он видоизменяется.