

ЛАБОРАТОРНАЯ РАБОТА № 7

ТЕМА: Свойство **style**. Объект **style** и его свойства

У всех HTML-элементов есть свойство **style**. Значением этого свойства является объект **style**, чьи собственные свойства позволяют изменять установки стилей.

Свойства объекта **style**

Свойства объекта **style** позволяет изменить стиль любого элемента Web-страницы, просто присвоив нужному свойству необходимое значение.

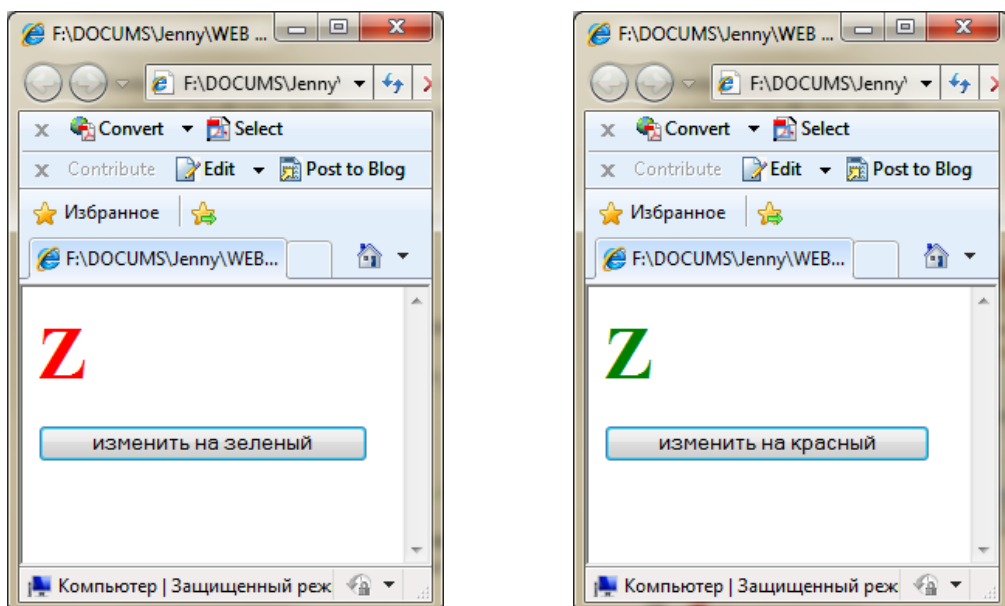
Существует четкое соответствие между свойствами стиля **CSS** и свойствами объекта **style** в **JavaScript**.

Свойства объекта **style** в **JavaScript** имеют почти такие же имена что и в **CSS** за тем исключением, что символы "-" убираются, а первые буквы всех слов, образующих имя атрибута, кроме первого, делаются прописными. В следующей таблице показаны примеры преобразования имен атрибутов стиля в имена свойств объекта **style**, устанавливающих стиль элемента.

css	JavaScript
background-attachment	backgroundAttachment
font-family	fontFamily
z-index	zIndex

Пример 1

По нажатию кнопки будем изменять цвет буквы и надпись на кнопке.



```

<head>
<style>
h1
{color:red; font-size:36pt;}
</style>
<script>
i=0;
function f()
{
// объект буква Z
r1=document.getElementById('e1')
// объект кнопка
r2=document.getElementById('e2')
if (i==0)
{
// у объекта r1 есть свойство style. Это тоже объект
// меняем у объекта style свойство color
r1.style.color='green';
// меняем у объекта r2 свойство value
r2.value='изменить на красный';
i=1;}
else
{
r1.style.color='red'; r2.value='изменить на зеленый';
i=0
}
}
</script>
</head>
<body>
<h1 id='e1'> Z </h1>
<form>
<input type="button" id='e2' value="изменить на зеленый"
onClick="f();">
</form>
</body>

```

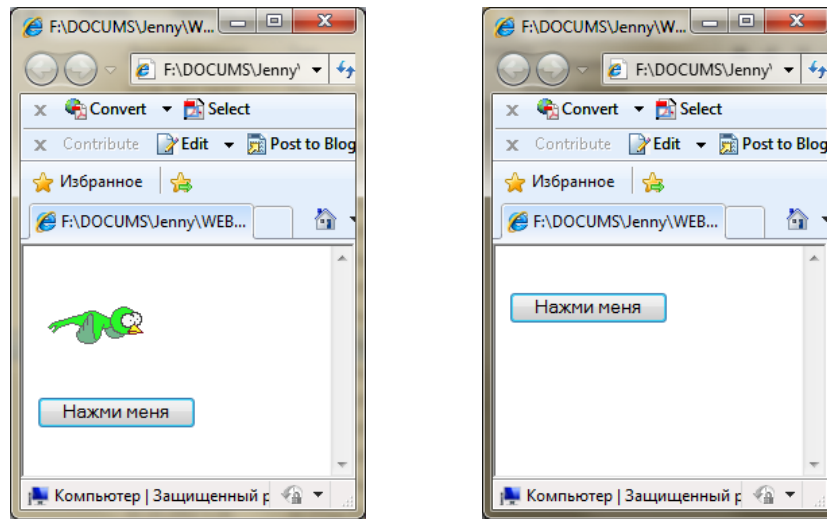
Свойство display

Свойство **display** задает способ отображения элемента на странице. Некоторые значения:

- **inline** — элемент ведет себя как линейный
- **block** — элемент ведет себя как блочный
- **none** — "удаляет" элемент со страницы вместе с содержимым. Элемент не видим, на странице блоки располагаются так, словно элемента нет

Пример 2

По нажатию кнопки картинка будет появляться и исчезать. Для этого будем пользоваться свойством **display** объекта **style**.



```
<head>
<script>
function displ()
{
p=document.getElementById('st')
// у объекта p есть свойство style. Это тоже объект
// меняем у объекта style свойство display
if (p.style.display == 'none')
{p.style.display = 'block'}
else {p.style.display = 'none'}
}
</script>
</head>
<body>

<br />
<input type="button" value="Нажми меня" onclick="displ();" >
</body>
```

Основы перемещения элементов

Движение объекта на **Web**-странице осуществляется путем изменения свойств, задающих его координаты. В **CSS** координаты объекта задаются с помощью свойств **left** и **top**. Свойство **left** задает горизонтальную координату объекта, которая в случае позиционирования в абсолютных координатах (**position: absolute**) задает расстояние в пикселях от левой границы экрана. Свойство **top** задает вертикальную координату объекта, которая в случае позиционирования в абсолютных координатах задает расстояние в пикселях от верхней границы экрана. Если задать элементу абсолютное позиционирование и изменять его координаты **top** и **left**, то элемент будет двигаться.

Метод setTimeout (expression, msec)

Метод setTimeout выполняет выражение или функцию по истечении установленного количества миллисекунд.

expression строковое выражение, содержащее имя вызываемой функции, msec числовое значение в миллисекундах. Выражение выполняется однократно и для его повторного выполнения требуется очередной вызов метода.

Пример 3

Создадим документ, в котором рисунок p5_0.jpg занимает 50% окна и является нижним слоем. На верхнем слое второй рисунок bird2.gif, который движется вправо. Когда второй рисунок достигает края первого рисунка, он останавливается.

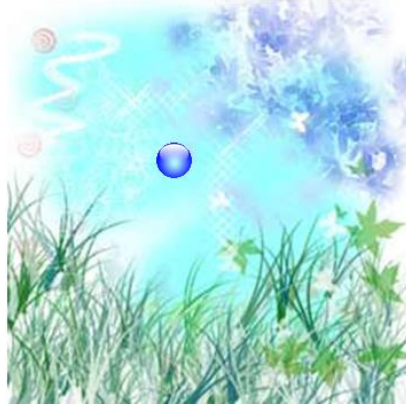


```
<head>
<style>
#im2 {position:absolute;left:0;top:100}
</style>
</head>
<body>


<script type="text/JavaScript">
function f()
{
// сдвиг рисунка на 10
x=x+10;
// изменяем свойство left 2-го рисунка
r2.style.left=x
// если он не дошел до края 1-го рисунка
// то вызываем каждые 1000 мили сек эту же функцию
if (x<x1-x2) setTimeout("f()",700);
}
// начало скрипта
x=0
r1=document.getElementById('im1')
r2=document.getElementById('im2')
x1=r1.width //ширина фона
x2=r2.width //ширина птички
f();
</script>
</body>
```

Пример 4

Изменим предыдущий документ так, чтобы рисунок 4.gif двигался по главной диагонали рисунка p5_0.jpg вниз. При достижении правого нижнего угла рисунок движется вверх по этой же диагонали. При достижении левого верхнего угла этот рисунок снова движется вниз. И так далее.



Шаг – расстояние слева и сверху от левого верхнего угла неподвижного рисунка до движущегося рисунка.

Если рисунок движется вправо и вниз, то шаг положительный. Если рисунок движется влево и вверх, то шаг отрицательный.

```
<html>
<head>
<style>
#im2 {position:absolute;left:0;top:0}
</style>
</head>
<body>

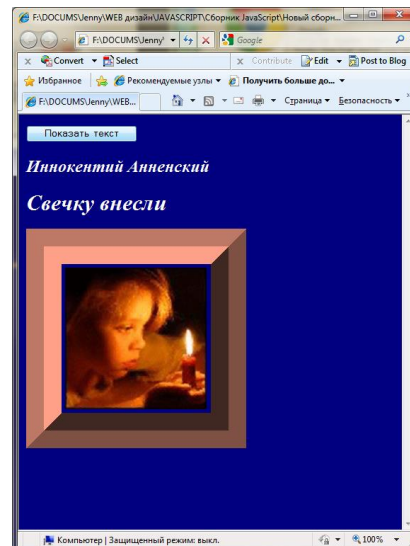
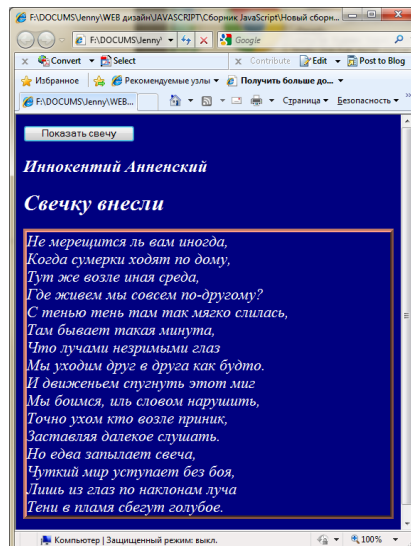

<script type="text/JavaScript">
function f()
{
// сдвиг рисунка на шаг s
x=x+s; y=y+s;
r2.style.left=x
r2.style.top=y
// если долетел до правого края меняем шаг на отрицательный
if (x>=x1) s=-10
// если долетел до левого края меняем шаг на положительный
if (x<=0) s=10
setTimeout("f()",50);
}
x=0 ; y=0; s=10;
r1=document.getElementById('im1')
r2=document.getElementById('im2')
x1=r1.width //ширина фона
f();
</script>
</body>
</html>
```

Вариант 1

1. Создайте документ, в котором рисунок `peznakomka3.jpg` заключен в рамку серого цвета типа `inset` и толщиной 50. По нажатию кнопки изменять цвет рамки на синий и надпись на кнопке. При следующем нажатии изменять цвет рамки на серый и надпись на кнопке.



2. Сначала на странице только текст стихотворения и кнопка с надписью «показать свечу». При нажатии на кнопку текст стихотворения прячется, на его месте появляется рисунок, и надпись на кнопке меняется на «показать текст». Если еще раз нажать на кнопку, рисунок исчезает и появляется текст. Текст стихотворения (Анненский.doc) и рисунок (свеча1.gif) скопировать из сетевой папки. Параметры форматирования страницы:
 - Цвет фона темно-синий, размер шрифта 22, курсив, цвет шрифта белый.
 - Рамка вокруг текста: цвет `#ffA089`, тип `outset`, толщина 5.
 - Картинка размером 200 на 200.
 - Рамка вокруг картинки: цвет `#ffA089`, тип `outset`, толщина 50, отступы по 5.



3. Создайте документ, в котором рисунок `p5_0.jpg` занимает 50% окна и является нижним слоем. На верхнем слое второй рисунок `4.gif`, который двигается по горизонтали к правой границе первого рисунка, затем на строку (высоту рисунка) вниз и назад, к левой границе, на строку вниз и к правой границе, и т.д. до нижней границы первого рисунка.



