

# ВВЕДЕНИЕ В JAVASCRIPT ДЛЯ МАГА

© 1996, 1997 *Стефан Кох (Stefan Koch)*

## Часть 9: Слои I

### Что такое слои?

Слои - это одна из замечательных новых возможностей браузера Netscape Navigator 4.0. Она позволяет выполнять точное позиционирование таких объектов web-страницы, как изображения. Кроме того, теперь Вы можете перемещать объекты по вашей HTML-странице. Вы можете также делать объекты невидимыми.

Управлять слоями можно легко с помощью языка JavaScript. Я надеюсь, что Вы, как и я, проникнитесь энтузиазмом в отношении возможностей этих слоев.

В данный момент пользоваться слоями Вы можете только в Netscape Navigator 4.0!

И, как обычно, я не буду детально описывать различные тэги. Хороший документ, описывающий все свойства такого объекта Netscape Navigator 4.0, как слои, находится по адресу <http://home.netscape.com/comprod/products/communicator/index.html> - и потому у меня нет нужды его пересказывать.

Что такое в действительности слои? Объяснять это очень легко на простом примере: берете несколько листов бумаги. На одном листе пишете текст. На другом - рисуете картинку. На третьем листе берете картинку и вписываете вокруг нее некий текст. И так далее. Теперь кладете эти листы на стол. Допустим, каждый лист - это некий слой. В этом смысле слой представляет собой в некотором роде контейнер. То есть он может включать в себя (содержать) некие объекты - в данном случае это будет текст и изображения.

Теперь берем бумагу с изображением и двигаем его по столу. Внимательно следите за тем, как это изображение движется вслед за бумагой. Если Вы сместим лист бумаги вправо, то и изображение тоже сдвинется! И что же собственно мы должны извлечь из этого увлекательного опыта? Слои, способные содержать различные объекты, например изображения, формы, текст, могут быть наложены на Вашу HTML-страницу и даже перемещаться по ней. Если Вы сдвигаете какой-либо слой, то и все содержащиеся в этом слое объекты тоже будут повторять это движение. Слои могут накладываться друг на друга подобно простым листам бумаги на столе. В каждом слое могут присутствовать прозрачные области. Сделайте в листе бумаги дырку. Теперь положите этот лист на другой. Такое отверстие - это 'прозрачная область' на первом листе - и через нее как раз видны нижележащие листы.

### Создание слоев

Чтобы создать слой, мы должны использовать либо тэг *<layer>* либо *<ilayer>*. Вы можете воспользоваться следующими параметрами:

Параметр	Описание
<code>name="layerName"</code>	Название слоя
<code>left=xPosition</code>	Абсцисса левого верхнего угла
<code>top=yPosition</code>	Ордината левого верхнего угла
<code>z-index=layerIndex</code>	Номер индекса для слоя
<code>width=layerWidth</code>	Ширина слоя в пикселах
<code>clip="x1_offset,y1_offset,x2_offset,y2_offset"</code>	Задаёт видимую область слоя
<code>above="layerName"</code>	Определяет, какой слой окажется под нашим
<code>below="layerName"</code>	Определяется, какой слой окажется над нашим
<code>Visibility=show hide inherit</code>	Видимость этого слоя
<code>bgcolor="rgbColor"</code>	Цвет фона - либо название стандартного цвета, либо rgb-запись
<code>background="imageURL"</code>	Фоновая картинка

Тэг *<layer>* используется для тех слоев, которые Вы можете точно позиционировать. Если же Вы не указываете положение слоя (с помощью параметров *left* и *top*), то по умолчанию он помещается в верхний левый угол окна.

Тэг *<ilayer>* создаёт слой, положение которого определяется при формировании документа.

Давайте теперь начнем с простого примера. Мы хотим создать два слоя. В первом из них мы помещаем изображение, а во втором - текст. Все, что мы хотим сделать - показать этот текст поверх данного изображения.



*Текст поверх изображения*

Исходный код:

```
<html>

<layer name=pic z-index=0 left=200 top=100>

</layer>

<layer name=txt z-index=1 left=200 top=100>
<font size=+4> <i> Layers-Demo </i> </font>
</layer>

</html>
```

Как видим, с помощью тэга `<layer>` мы формируем два слоя. Оба слоя позиционируются как 200/100 (через параметры *left* и *top*). Все, что находится между тэгами `<layer>` и `</layer>` (или тэгами `<ilayer>` и `</ilayer>`) принадлежит описываемому слою.

Кроме того, мы используем параметр *z-index*, определяя тем самым порядок появления указанных слоев - то есть, в нашем случае, Вы тем самым сообщаете браузеру, что текст будет написан поверх изображения. В общем случае, именно слой с самым высоким номером *z-index* будет показан поверх всех остальных. Вы не ограничены в выборе *z-index* лишь значениями 0 и 1 - можно выбирать вообще любое положительное число.

Так, если в первом тэге `<layer>` Вы напишете *z-index=100*, то текст окажется под изображением - его слой номер Z-индекса (*z-index=1*). Вы сможете увидеть текст сквозь изображение, поскольку я использовал в нем прозрачный фон (формат gif89a).



*Текст под изображением*

## Слои и JavaScript

Рассмотрим теперь, как можно получить доступ к слоям через JavaScript. Начнем же мы с примера, где пользователь получает возможность, нажимая кнопку, прятать или показывать некий слой.

Для начала мы должны знать, каким образом слои представлены в JavaScript. Как обычно, для этого имеются несколько способов. Самое лучшее - дать каждому слою свое имя. Так, если мы задаем слой

```
<layer ... name=myLayer>
```

```
...
```

```
</layer>
```

то в дальнейшем можем получить доступ к нему с помощью конструкции `document.layers["myLayer"]`. Согласно документации, предоставляемой фирмой Netscape, мы можем также использовать запись `document.myLayer` - однако в моем браузере это приводит к сбою. Конечно, это всего лишь проблема предварительной версии и в заключительном варианте будет успешно решена (сейчас я пользуюсь Netscape Navigator 4.0 PR3 на WinNT). Однако, по-видимому, нет никаких проблем с конструкцией `document.layers["myLayer"]` - поэтому мы и будем пользоваться именно такой альтернативой из всех возможных.

Доступ к этим слоям можно также получить через целочисленный индекс. Так, чтобы получить доступ к самому нижнему слою, Вы можете написать `document.layers[0]`. Обратите внимание, что индекс - это **не** то же самое, что параметр *z-index*. Если,

например, Вы имеете два слоя, называемые layer1 и layer2 с номерами z-index 17 и 100, то Вы можете получить доступ к этим слоям через `document.layers[0]` и `document.layers[1]`, а **не** через `document.layers[17]` и `document.layers[100]`.

Слои имеют несколько свойств, которые можно изменять с помощью скрипта на JavaScript. В следующем примере представлена кнопка, которая позволяет Вам скрывать или, наоборот, предоставлять один слой (требуется Netscape Navigator версии 4.0 или выше).

*(online-версия руководства позволит Вам проверить этот скрипт немедленно)*

Исходный код скрипта выглядит следующим образом:

```
<html>
<head>
<script language="JavaScript">
<!-- hide

function showHide() {
  if (document.layers["myLayer"].visibility == "show")
    document.layers["myLayer"].visibility= "hide"
  else document.layers["myLayer"].visibility= "show";
}

// -->
</script>
</head>
<body>

<ilayer name=myLayer visibility=show>
<font size=+1 color="#0000ff"><i>This text is inside a layer</i></font>
</ilayer>

<form>
<input type="button" value="Show/Hide layer" onClick="showHide()">
</form>

</body>
</html>
```

Данная кнопка вызывает функцию `showHide()`. Можно видеть, что в этих функциях реализуется доступ к такому свойству объекта layer (`myLayer`), как видимость. Присвоивая параметру `document.layers["myLayer"].visibility` значения `"show"` или `"hide"`, Вы можете показать или скрыть наш слой. Заметим, что `"show"` и `"hide"` - это строки, а не зарезервированные ключевые слова, то есть Вы **не можете** написать `document.layers["myLayer"].visibility= show`. Вместо тэга `<layer>` я также пользовался тэгом `<ilayer>`, поскольку хотел поместить этот слой в "информационный поток" документа.

## Перемещение слоев

Свойства `left` и `top` определяют положение данного слоя. Вы можете менять его, записывая в эти атрибуты новые значения. Например, в следующей строке задается горизонтальное положение слоя в 200 пикселей:

```
document.layers["myLayer2"].left= 200;
```

Перейдем теперь к программе перемещения слоев - она создает нечто вроде линейки прокрутки внутри окна браузера.

*(online-версия руководства позволит Вам проверить этот скрипт немедленно)*

Сам скрипт выглядит следующим образом:

```
<html>
<head>
<script language="JavaScript">
<!-- hide

function move() {
  if (pos < 0) direction= true;
  if (pos > 200) direction= false;

  if (direction) pos++
  else pos--;

  document.layers["myLayer2"].left= pos;
}

// -->
</script>
</head>
<body onLoad="setInterval('move()', 20)">

<ilayer name=myLayer2 left=0>
<font size=+1 color="#0000ff"><i>This text is inside a layer</i></font>
</ilayer>

</body>
</html>
```

Мы создаем слой с именем *myLayer2*. Можно видеть, что в тэге `<body>` мы пользуемся процедурой `onLoad`. Нам необходимо начать прокручивание слоя, как только страница будет загружена. В процедуре обработки события `onLoad` мы пользуемся функцией `setInterval()`. Это один из новых методов версии 1.2 языка JavaScript (то есть версии JavaScript, реализованной в Netscape Navigator 4.0). Им можно пользоваться, чтобы вызывать некую функцию многократно и вновь через определенные интервалы времени. В прошлом для этого мы пользовались функцией `setTimeout()`. Функция `setInterval()` работает почти так же, однако Вам нужно вызвать ее всего лишь один раз.

С помощью *setInterval()* мы вызываем функцию *move()* каждые 20 миллисекунд. А функция *move()*, в свою очередь, всякий раз смещает слой на новую позицию. И поскольку мы вызываем эту функцию вновь и вновь, то мы получаем быстрый скроллинг нашего текста. Все, что мы нужно сделать в функции *move()* - это вычислить новую координату для слоя и записать ее: *document.layers["myLayer2"].left= pos*.

Если Вы посмотрите исходный код этой части в онлайн-описании, то увидите, что в действительности мой код выглядит несколько иначе - я добавил некий фрагмент кода с тем, чтобы люди, работающие со старыми версиями JavaScript-браузеров, не получали из-за этого никаких сообщений об ошибках. Как этого можно достичь? Следующий фрагмент кода будет выполняться только на тех браузерах, которые воспринимают язык JavaScript 1.2:

```
<script language="JavaScript1.2">
<!-- hide
document.write("You are using a JavaScript 1.2 capable browser.");
// -->
</script>
```

Та же самая проблема возникает, когда мы работаем с объектом Image. Мы можем аналогичным способом переписать кодексы. Установка переменной *browserOK* решает эту проблему.

*(online-версия руководства позволит Вам проверить этот скрипт немедленно)*

©1996,1997 by Stefan Koch  
e-mail:skoch@rumms.uni-mannheim.de  
<http://rummelplatz.uni-mannheim.de/~skoch/>  
Моя книга по JavaScript: <http://www.dpunkt.de/javascript>