

Визуальные эффекты. Управление видимостью и переполнением

С помощью особых свойств языка CSS2 могут быть сработаны некоторые визуальные эффекты, такие как: обработка переполнения и управление видимостью.

Переполнение.

Свойство overflow

Бывают случаи, когда содержимое структурного блока не помещается в его информативной области. При этом возникает

ситуация, называемая переполнением. Такое явление может возникнуть, когда:

- явно заданные геометрические размеры информативной области (значения width и height) не позволяют вместить в нее требуемую информацию в том виде, в каком она задана. Например: для абзаца отведена область 4x5 см, а на нее надо поместить страницу печатного текста;
- структурный блок является абсолютно позиционируемым и выходит за рамки своего контейнера.

В первом случае по умолчанию размеры блока будут увеличены до значений, необходимых для полного вывода всей содержащейся

в нем информации. Второй случай зависит от браузера. Обычно ничто не мешает абсолютно позиционируемым блокам

отображаться вне своих контейнеров, назначенных их родителями. Это верно до тех пор, пока абсолютно позиционируемый

блок находится в рамках начального контейнера. Если такой блок выходит за пределы начального контейнера, то он усекается.

Свойством CSS2, отвечающим за обработку ситуации переполнения, является свойство overflow. Через него задается, что

будет происходить с непомещающейся в структурный блок информацией. Свойство overflow может принимать следующие значения:

- visible;
- hidden;
- scroll;
- auto.

рассмотрим каждые из них в отдельности.

Visible это значение говорит о том, что вся информация (и дочерние блоки

в том числе), заключенная в структурном блоке должна быть отображена. Причем отображена в пределах структурного блока.

Это приводит к увеличению размеров блока до необходимого значения. Значение VISIBLE используется по умолчанию (рис. 5.35).

значении visible

Пример использования (рис. 5.36)

```
<html>

<head>

<title>
```

рис. 5.35.

Принцип работы свойства overflow в

Демонстрация использования CSS-свойства overflow </title>

```
<style type="text/css"> p { border:double black; overflow:visible; width:200}
```

```
body {text-align:center} </style> </head> <body>
```

```
<h3>Первоначальная ширина блока равна 200. Свойство overflow : visible</h3> <p>
```

```
 </p>
```

```
</body>
```

```
</html>
```



рис. 5.36. результат применения свойства overflow в значении visible

Hidden..... это значение указывает на то, что вся информация (и дочерние

блоки в том числе), не уместившаяся в структурном блоке заданных размеров, будет усекаться и отображена не будет (рис.

5.37).

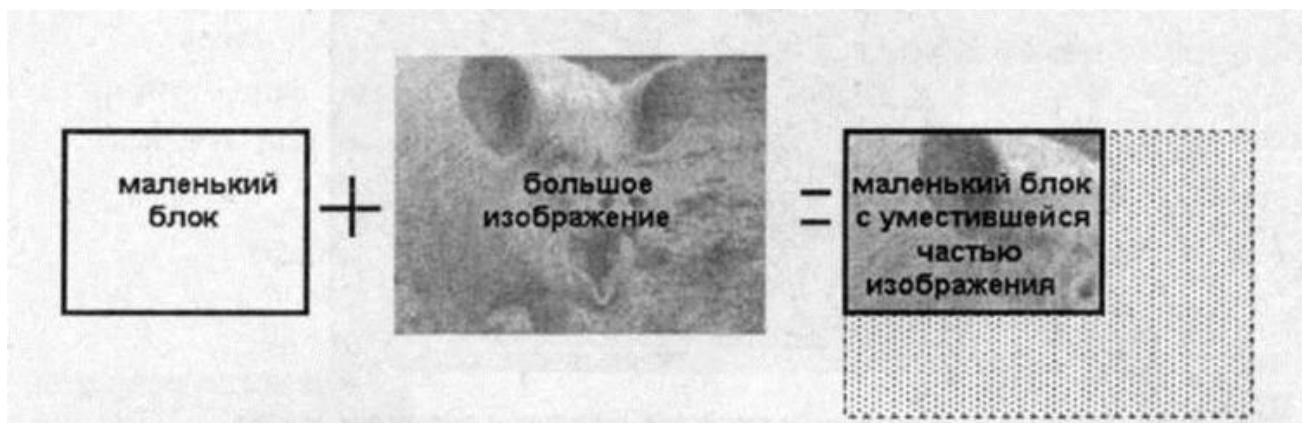


рис. 5.37. Принцип работы свойства overflow в значении hidden

Пример использования (рис. 5.38)

```
<html>

<head>

<title>
Демонстрация использования CSS-свойства overflow </title>

<style type = "text/css"> p { border:double black; overflow:hidden; width:200}
```

```

body {text-align:center}

</style>

</head>

<body>

<h3>Первоначальная ширина блока равна 200. Свойство overflow : hidden</h3>

<p>

 </p> </body>

</html>

```

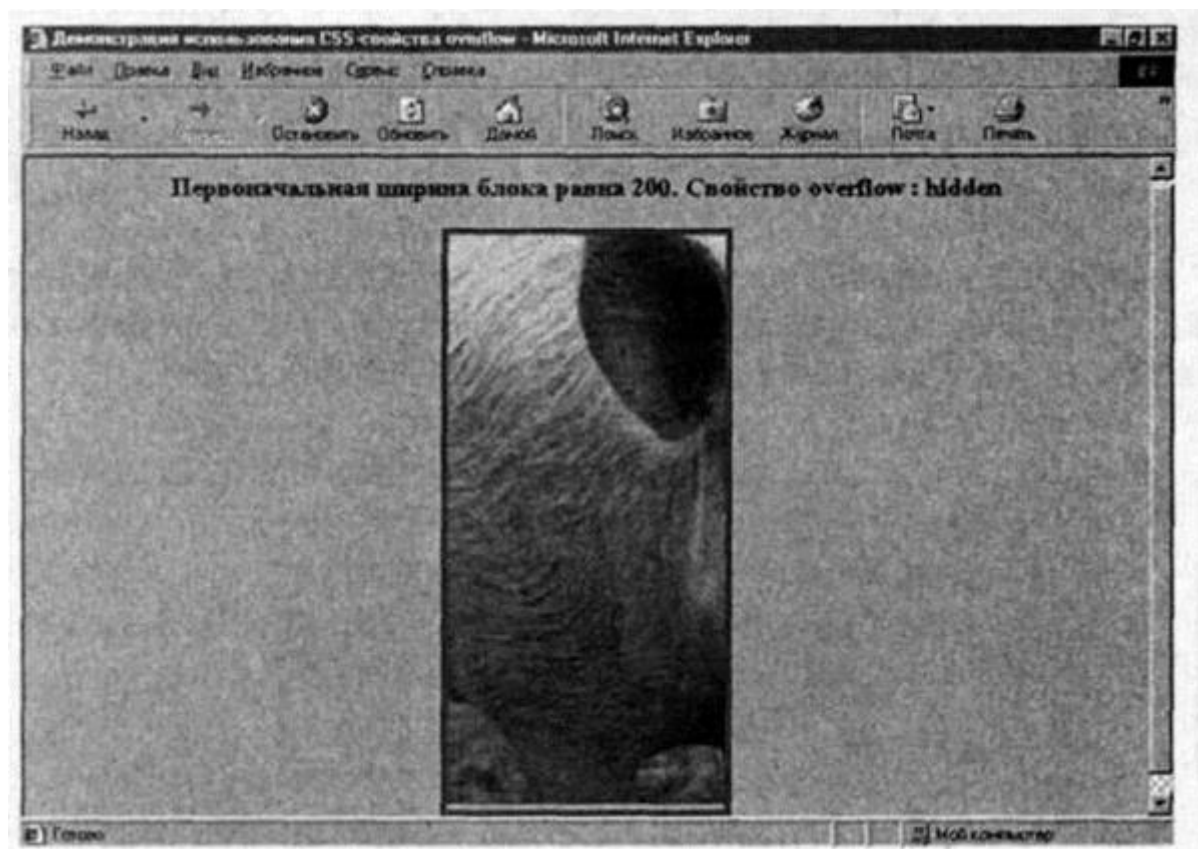


рис. 5.38. результат применения свойства overflow в значении hidden

Scroll использование этого ключевого слова в качестве значения свойства

overflow позволяет установить полосы прокрутки, чем достигается возможность доступа к информации без изменения размеров

пространства, занимаемого структурным блоком на странице.

Auto обработка возникшей ситуации переполнения определяется браузером. Им может быть использовано одно из

вышеописанных значений в зависимости от контекста ситуации и своих индивидуальных настроек. Обычно при значении auto в

случаях переполнения браузер добавляет полосу прокрутки. Причем он может добавить только одну полосу

прокрутки, тогда

как при значении scroll всегда добавляется обе полосы, даже если одна из них не нужна (рис. 5.39).

Управление видимостью

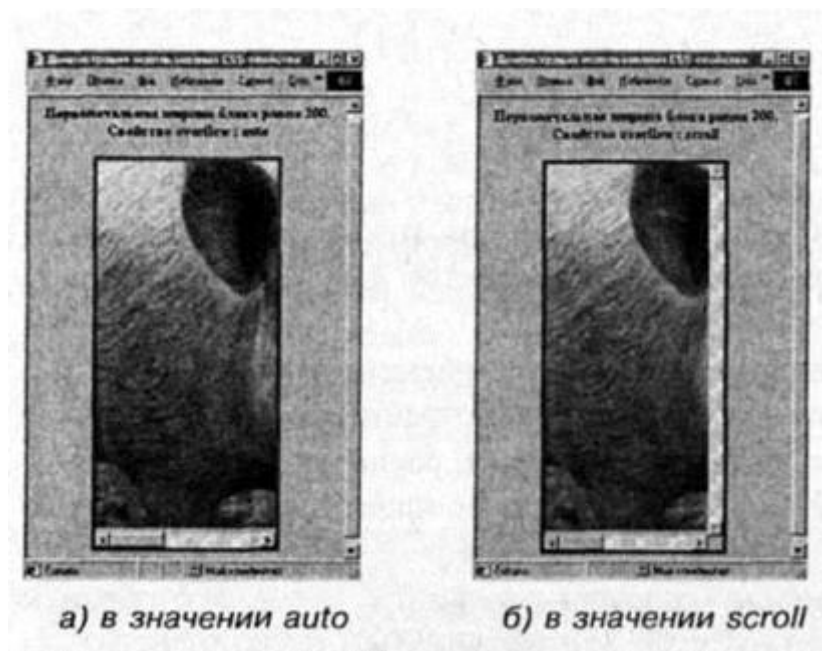
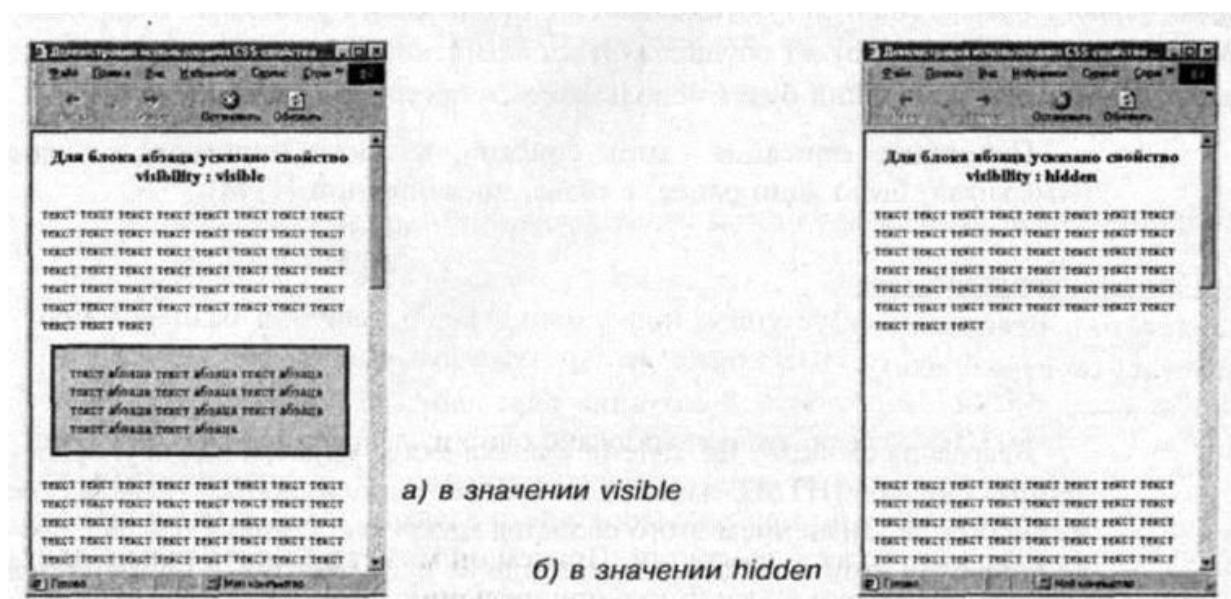


рис. 5.39. результат применения

С помощью CSS-свойства может быть задана видимость структурного блока, то есть, виден или не виден будет блок пользователю. Таким CSS-свойством является свойство

visibility. В соответствии со своим назначением оно может принимать следующие значения: visible — при этом

структурный блок является видимым.



hidden — при таком значении весь структурный блок становится невидимым и не отображается. Все остальные блоки ведут себя

так, как будто его нет. Свойство visibility поддерживается браузером Netscape Communicator только 6 версии. Netscape

рис. 5.40. Демонстрация использования CSS-свойства visibility

Communicator 4.74 его игнорирует (рис. 5.40).

Отображение списков средствами языка CSS2

Возможности языка CSS2 позволяют и для списков задавать стилевые настройки. При этом для html-элементов UL и OL могут

быть определены следующие свойства:

list-style-type..... задает тип маркера или нумерацию элементов списка. Это тип

выбирается из стандартных, предусмотренных в CSS2 маркеров;

list-style-image... позволяет использовать изображения, располагающиеся во внешних

графических файлах, в качестве маркеров (по умолчанию имеет значение none);

list-style-position.... указывает положение маркера в списке, т.е. будет он отображен в

составе элемента списка или выдвинутым влево от него.

Свойство list-style-type определяет тип маркера или нумерацию элементов списка в том случае, если для этих целей не

используются изображения (list-style-image: none), или они по каким-либо причинам недоступны (например, неправильно

указан их URL-адрес). В качестве значений для свойства list-style-type могут применяться следующие ключевые слова:

none..... вообще никакой маркер не отображается (значение по умолчанию);

disk..... в качестве маркера используется закрашенный кружок;

circle..... в качестве маркера используется не закрашенный кружок (окружность);

5

square..... маркер имеет вид закрашенного квадрата;

decimal сдает нумерацию списка арабскими цифрами;

lower-roman... задает нумерацию списка строчными римскими цифрами;

upper-roman .. нумерация будет осуществляться прописными римскими цифрами;

lower-alpha.... нумерация будет осуществляться латинскими строчными буквами;

upper-alpha.... для нумерации будут использоваться прописные латинские буквы. Подробное описание самих списков, а также маркеров и видов нумераций, было дано ранее, в главе, посвященной html.

Пример использования:

```
UL {color: green; list-style-type: disk  
}
```

Благодаря свойству list-style-image в качестве маркера для неупорядоченных списков (html-элемент UL) может использоваться графическое изображение. Значением этого свойства может быть либо ключевое слово none (при этом никакое изображение не используется, а применяется один из стандартных маркеров), либо URL-адрес файла, в котором изображение хранится.

Пример задания:

```
UL ( list-style-image: url (http://myserver.da.ru/pulka.gif) }
```

В свою очередь, свойство list-style-position задает положение маркера относительно элемента списка. При этом возможны два варианта:

Inside..... как бы встраивается в элемент списка и является его первым символом;

outside..... маркер отображается несколько смещенным влево относительно элемента списка.

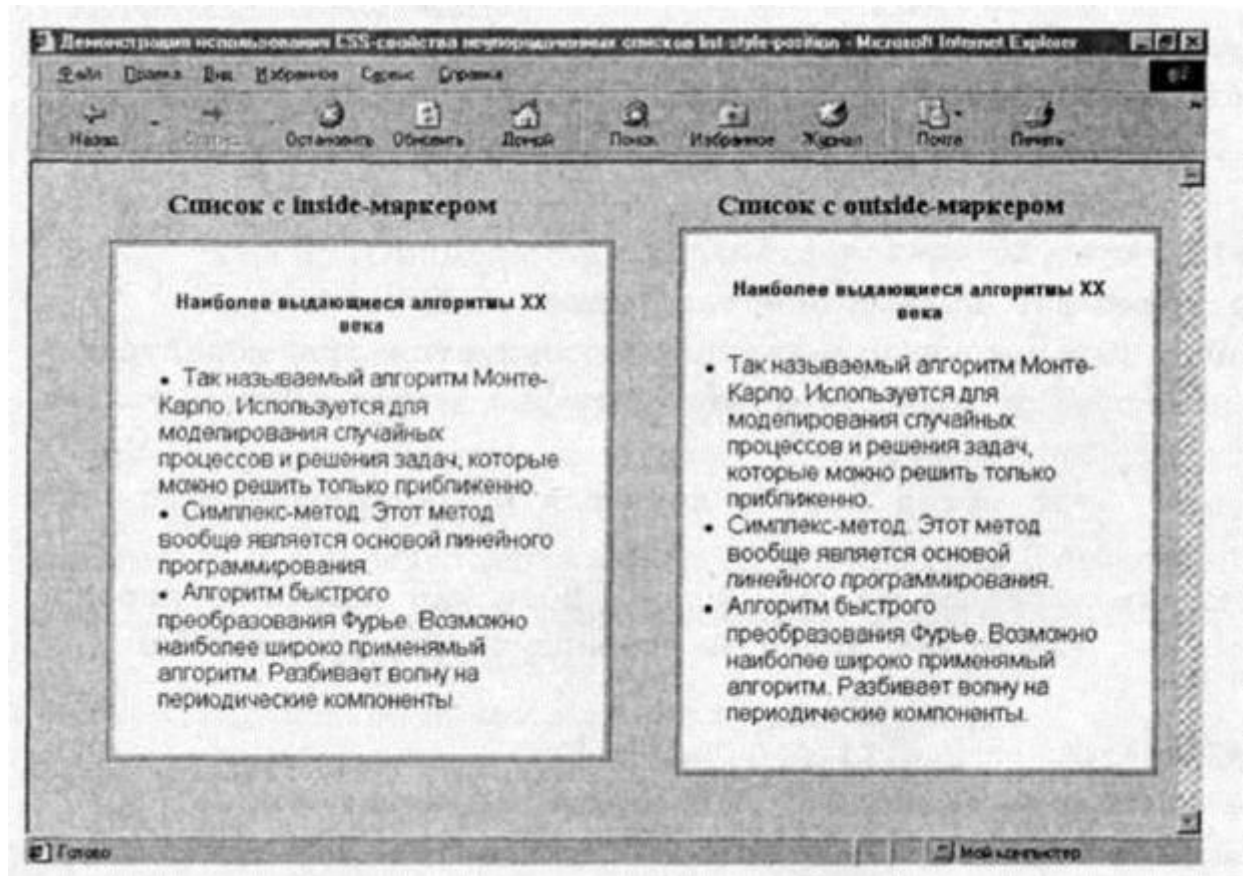


рис. 5.41. Демонстрация использования свойства list-style-position

Наглядно разницу между ними можно увидеть на следующем примере (рис. 5.41).

```

<html>

<head>

<title>
Демонстрация .использования CSS-свойства неупорядоченных списков list-style-position </title>

<style type ="text/css"> UL { padding:30; border: double gray;font-family:"Arial";
background-color:white} body {text-align:center; background-color:lightgrey( th { font-
size:14pt}

</style>

</head>

<body>

<TABLE>

<tr>

<TH>Список с inside-маркером

<TH>Список с outside-маркером

<tr>

<TD>

<UL style="list-style-position:inside ">

<H5 align=center>Наиболее выдающиеся алгоритмы XX века</H5>

<LI>Так называемый алгоритм Монте-Карло. Используется для моделирования случайных процессов и
решения задач, которые можно решить только приближенно.

<LI>Симплекс-метод. Этот метод вообще является основой линейного программирования .

<LI>Алгоритм быстрого преобразования Фурье. Возможно, наиболее широко применяемый алгоритм.
разбивает волну на периодические компоненты. <TD>

<UL style="list-style-position:outside ">

<H5 align=center>Наиболее выдающиеся алгоритмы XX века</H5>

<LI>Так называемый алгоритм Монте-Карло. Используется для моделирования случайных процессов и
решения задач, которые можно решить только приближенно.

<LI>Симплекс-метод. Этот метод вообще является основой линейного программирования .

<LI>Алгоритм быстрого преобразования Фурье. Возможно, наиболее широко применяемый алгоритм.
разбивает волну на периодические компоненты. </TABLE>

</body>

</html>

```


Задание формы курсора.

Свойство cursor

CSS-свойство cursor позволяет разработчику задавать для курсора свой авторский внешний вид. В качестве значения этого

свойства указывается URL-адрес файла подключаемого курсора, причем задаваться может целый список URL-адресов, каждый из

которых будет опрашиваться в порядке очередности, если с предыдущего URL-адреса подгрузить курсор не удастся.

Пример использования:

```
body {cursor: url ("myfirst.cur"), url ("mysecond.cur") }
```

В этом примере сначала будет попытка подключить курсор из файла myfirst.cur и если она окажется безуспешной, будет

подключаться второй курсор. Примечательно, что форму курсора можно изменять в области каждого html-элемента индивидуально. Например, можно изменять его форму, если он находится под абзацем: p {cursor: url ("coolcur.cur")}

В качестве ключевых слов могут быть использованы следующие:

default устанавливается курсор, используемый по умолчанию (обычно в виде стрелки);

crosshair..... устанавливается курсор в виде крестика (пересечения двух отрезков);

text такой вид курсор принимает, когда указывает на текст, который можно выделить;

wait курсор принимает вид песочных часов;

auto вид курсора задается браузером автоматически от контекста.