

### ТЕМА 3.3

## ТЕХНОЛОГИИ ВИЗУАЛЬНОГО ПРЕДСТАВЛЕНИЯ ДОКУМЕНТА. БЛОКОВАЯ СТРУКТУРА. ПОНЯТИЕ О КОНТЕЙНЕРЕ

Нормальный поток

Схемы позиционирования.

Абсолютное позиционирование

Эти расстояния задаются с помощью CSS-свойств:

Относительное позиционирование

Перемещаемые блоки. Свойство float

Многослойный вывод. Свойство Z-index

## ТЕМА 3.3

# ТЕХНОЛОГИИ ВИЗУАЛЬНОГО ПРЕДСТАВЛЕНИЯ ДОКУМЕНТА. БЛОКОВАЯ СТРУКТУРА. ПОНЯТИЕ О КОНТЕЙНЕРЕ

Как было написано в одной из предыдущих глав, под блокообразующие html-элементы при выводе документа отводятся стилевые блоки, обладающие определенными визуальными свойствами (отступы, границы, поля). Под строковые html-элементы такие блоки не выделяются. Все строковые элементы входят в состав блокообразующих элементов и отображаются в их информативной области. Блокообразующим элементом самого высокого уровня является элемент body.

С точки зрения структуры использование блоков выглядит не совсем так. В CSS2 каждый html-элемент, присутствующий в дереве документа (элемент BR, например, не присутствует), помещается в структурный блок. Структурные блоки блокообразующих html-элементов называются главными. Таким образом, только главные структурные блоки могут иметь свое, индивидуальное стилевое оформление (отступы, границы, поля). Именно они и их параметры, используются при форматировании (компоновке) документа. Главный блок может содержать либо структурную единицу текста (например, абзац), либо несколько структурных блоков, являющихся его потомками (например, несколько абзацев). Вообще, все структурные блоки, имеющие внутри себя хотя бы один дочерний структурный блок, ничего, кроме структурных блоков, содержать не могут. рассмотрим это положение на примере:

<p>

Текст абзаца Текст абзаца

<pRE>

Отформатированный текст абзаца Отформатированный текст абзаца Отформатированный текст абзаца

</pRE>

</p>

В этом примере главный структурный блок содержит в себе некий текст абзаца, а также дочерний структурный блок (тоже главный), порожденный элементом pRE .

В соответствии с вышеприведенным положением CSS2 этого быть не должно. Выход из создавшегося положения заключается в том, что под текст абзаца выделяется безымянный структурный блок. И, таким образом, главный структурный блок, порожденный элементом p,

становится обладателем двух дочерних структурных блоков и ничего более чем и достигается соблюдение требований языка CSS2 .

Строковые элементы, не образующие новых структурных единиц текста, заключаются в строковые структурные блоки. Например:

<p>

Пример армейского юмора:

<STRONG> Приказ командира </STRONG>

<EM> Всем отсутствующим построиться в отдельную шеренгу </EM>

</p>

Элемент p порождает главный структурный блок, который содержит несколько строковых структурных блоков:

• безымянный строковый блок ("Пример армейского юмора");

• строковый структурный блок, порожденный элементом STRONG ("Приказ командира");

• строковый структурный блок, порожденный элементом EM ("Всем отсутствующим построиться в отдельную шеренгу").

Такая блоковая структура позволяет наиболее полно и просто представлять структуру документа в рамках его блочной модели. Что касается практики, то на первый взгляд может показаться, что достаточно было бы главных блоков, обладающих индивидуальным стилевым оформлением (поля, границы, отступы). Однако помимо декоративного оформления, благодаря блоковой организации всего документа осуществляется задание месторасположения (позиционирование) отдельных фрагментов информации в пределах документа. Это означает, что, например, положение фрагмента текста на Web-странице задается положением структурного блока, которому он принадлежит. Причем этим фрагментом текста может быть как структурное подразделение текста (абзац), так и отдельная его строка или даже слово.

В CSS2 положение и размеры блоков определяются относительно краев определенной области прямоугольной формы. Эта область называется контейнером. Контейнером для любого структурного блока является пространство, отведенное для информативной области его родительского структурного блока. При этом говорят, что родительский блок назначает контейнер своему потомку.

Стоит отметить, что относительно контейнера задается положение дочернего блока, но размеры последнего никоим образом не ограничиваются размерами контейнера, назначенного родительским блоком. Другими словами, структурный блок может выходить за рамки своего контейнера.

Ввиду всего вышесказанного становится понятным, что начальный контейнер назначается корневым html-элементом дерева документа, т.е. либо элементом `body`, либо элементом `FRAMESET`. Именно начальный контейнер является контейнером самого высокого уровня. Его нельзя позиционировать (задавать его положение в контексте других блоков) и перемещать. Это значит, что применительно к нему игнорируются соответствующие CSS2-свойства `position` и `float`, описываемые далее. Можно задавать только геометрические размеры начального контейнера, указав атрибуты `width` и `height` корневого html-элемента. Дочерние блоки всех поколений не могут выходить за рамки начального контейнера.

Теперь рассмотрим, где, как и в какой последовательности могут отображаться структурные блоки в рамках визуального представления документа. Если язык html позволял осуществлять только последовательный вывод (содержимое элементов располагается на странице друг за другом в том порядке, в котором заданы сами html-элементы), то язык CSS2 позволяет реализовать более широкие возможности позиционирования.

## Нормальный поток

В этом разделе будет произведено описание потока или, другими словами, последовательности вывода содержимого html-документа, являющейся обычной и используемой по умолчанию. Именно нормальный поток вывода осуществляется обычными средствами языка html. В этом случае информация отображается в том же порядке, в котором она задана в html-коде документа.

рассмотрим нормальный поток через призму языка CSS2 и его блочную организацию документа. При нормальном потоке структурные блоки располагаются друг за другом вертикально, начиная с верхнего края своего контейнера, назначенного родительским блоком, в порядке, в котором они указаны в html-документе. Вертикальное расстояние между двумя сестринскими блоками определяется размером их полей (свойство `margin`). Причем поля соседних блоков перекрываются и за расстояние между последними принимается размер наибольшего поля. Если по вертикали соприкасаются главный и строковый структурный блоки, то расстояние между ними определяется размером поля главного блока, т.к. строковый блок вообще не имеет поля. Левое поле каждого главного блока или левый край строкового блока соприкасается с левым краем своего контейнера. Наглядно это можно увидеть на рисунке 5.26.

Если имеет место направленность текста справа налево, то соприкасаются правые поля и края. Строковый блок выводится одной строкой.

Шириной строкового блока является ширина назначенного ему контейнера. Несколько идущих подряд строковых структурных блоков также выводятся одной строкой.

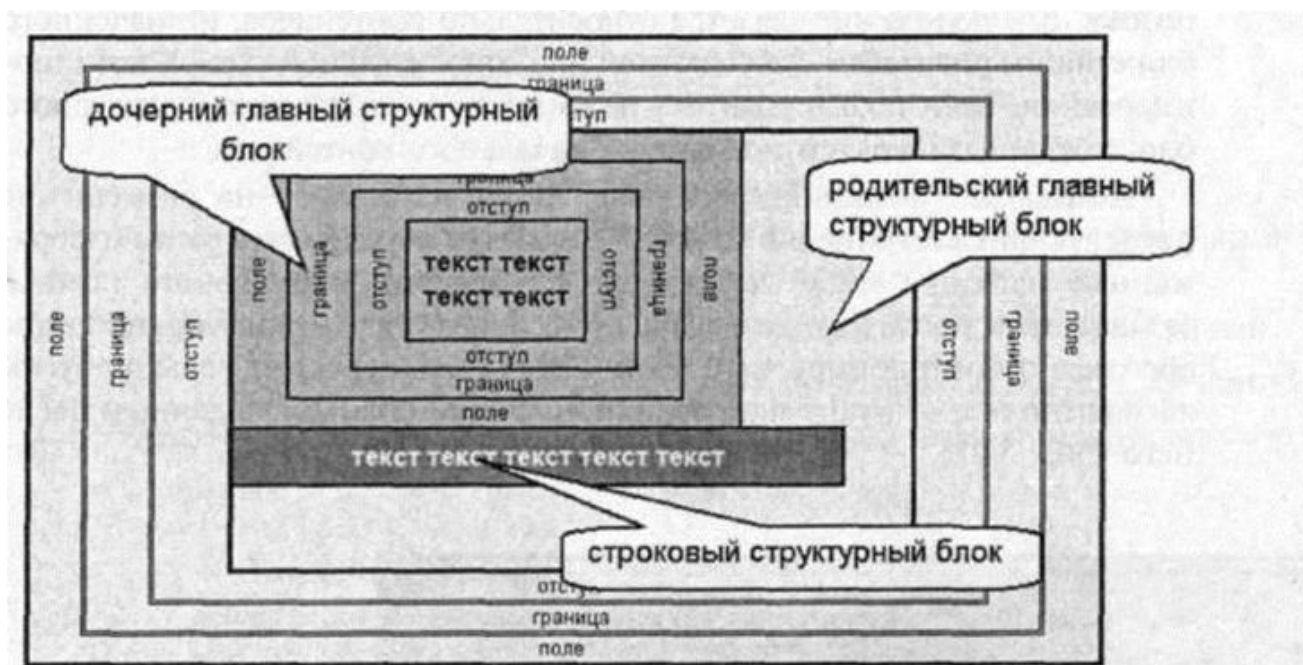


рис. 5.26. Пример отображения содержимого элементов в нормальном потоке.

## Схемы позиционирования.

Свойство position\

Средствами языка CSS2 структурный блок, может быть, вынут из нормального потока отображения документа. В этом случае его месторасположение на Web-странице перестает зависеть от его положения в очередности задания блоков, то есть не зависит от того, в каком месте html-кода документа задан элемент. Теперь его визуальное расположение на странице может быть задано независимо от других блоков. Причем он может перекрывать ранее отображенные блоки. Такая схема позиционирования называется абсолютной.

К тому же положение блока может быть задано относительно его обычного положения в нормальном потоке. В этом случае будет иметь место схема относительного позиционирования.

Схема, соответствующая нормальному потоку, называется схемой статического позиционирования. Задание схемы, согласно которой будет позиционироваться блок, осуществляется свойством position. В соответствие со своим назначением оно может принимать следующие значения:

static\ — указывает на то, что текущий блок является обычным, располагающимся в соответствии с нормальным потоком блоков, другими словами, задает статическую схему позиционирования для данного блока;

relative\ — указывает относительную схему позиционирования для данного блока;

absolute\ — задает абсолютную схему позиционирования для данного блока.

## Абсолютное позиционирование

Структурный блок, для которого указана абсолютная схема позиционирования, изымается вместе со всеми своими потомками из нормального потока. Его положение задается относительно контейнера, назначенного ближайшим родителем с абсолютной

или относительной схемой позиционирования. Если такого родителя не имеется, то положение структурного блока будет задаваться относительно начального контейнера.

Абсолютно позиционированные блоки не влияют на размещение последующих сестринских блоков. Они могут перекрывать ранее отображенные блоки, а также перекрываться блоками нормального потока, задаваемыми после них. Это значит, что при отображении Web-страницы, абсолютно позиционируемый блок может быть перекрыт только другим абсолютно или относительно позиционируемым блоком, заданным после него (рис. 5.27).

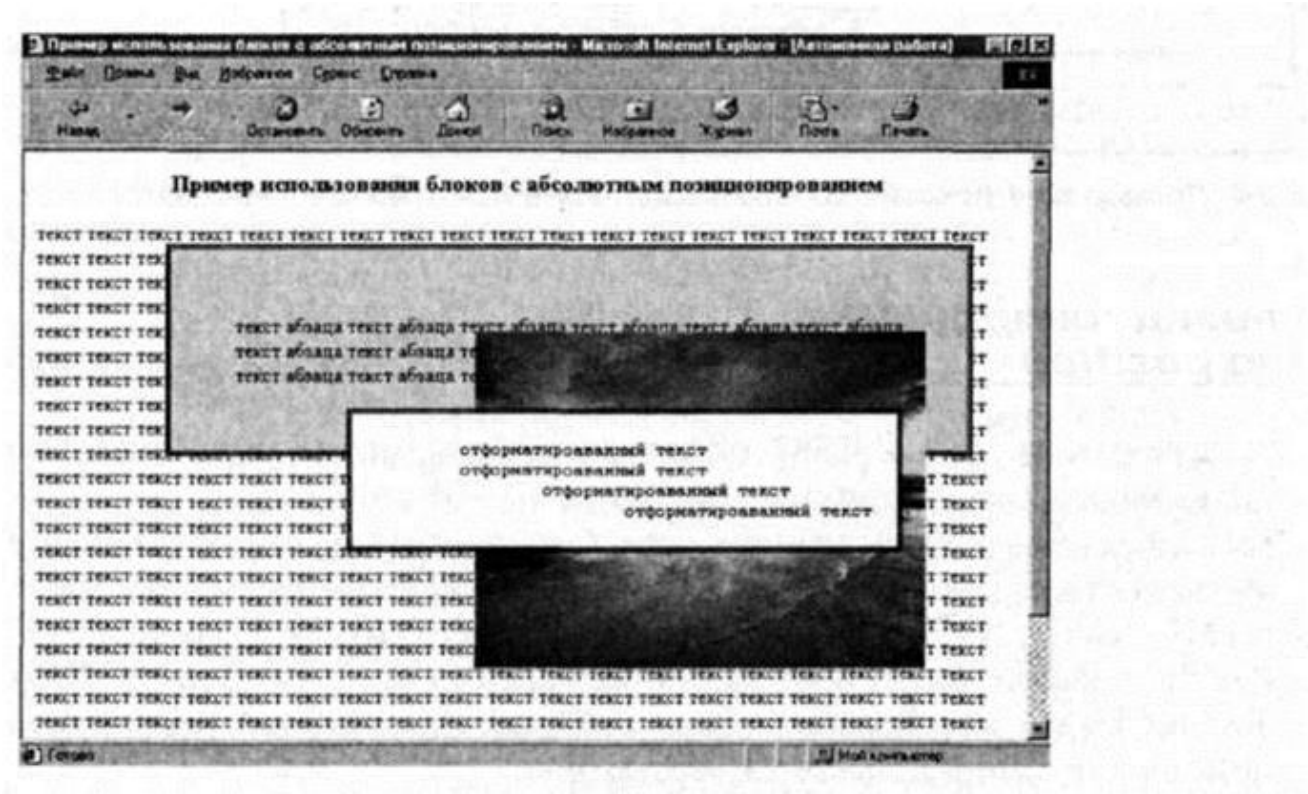


рис. 5.27. Пример отображения абсолютно позиционируемых блоков и демонстрация наложения их друг на друга

Надо также отметить, что поля, абсолютно позиционируемых главных блоков не перекрываются никакими другими полями. Положение такого блока задается:

- по вертикали — расстоянием между верхним краем контейнера и верхним краем структурного блока или расстоянием между нижними краями;
- по горизонтали — расстоянием между левым краем контейнера и левым краем структурного блока или расстоянием между правыми краями. Например:

```
img {position : absolute; top: 30px; left: 100px}
```

# Эти расстояния задаются с помощью CSS-свойств:

top — своим значением указывает величину смещения верхнего края блока относительно верхнего края контейнера;

bottom — задает расстояние между нижними краями блока и контейнера;

left — задает смещение левого края блока относительно левого края контейнера;

right — задает смещение правого края блока относительно правого края контейнера.

Значения этих свойств задаются в стандартных единицах длины, определенных для языка CSS2. Также может быть использовано ключевое слово auto. Использование этого слова в качестве значения для свойства left приведет к тому, что левый край блока будет совпадать с левым краем контейнера. Если свойство top выставлено в значении auto, то совпадать будут верхние края блока и контейнера. Свойства right и bottom, прошу обратить на это особое внимание, не поддерживаются браузером Netscape Communicator, поэтому их использование является нежелательным. Для абсолютного позиционирования применяйте только свойства top и left. Их вполне достаточно. По умолчанию оба этих атрибута используются в значении auto.

Использование свойств right и bottom для браузера Internet Explorer также имеет свои особенности:

Ø если для одного абсолютно позиционируемого блока указаны два свойства left и right, и оба в числовом значении, то свойство left перекрывает свойство right независимо от порядка следования;

Ø если оба свойства right и left выставлены в значении auto, то свойство left опять же перекрывает свойство right;

Ø если для одного из них указано значение auto, а для другого числовое значение, то последнее свойство перекрывает свойство со значением auto.

Совместное применение свойств top и bottom аналогично вышеописанному применению left и right. Только вместо left надо подставить top, а вместо right — bottom. В качестве примера приведен код документа, отображенного на рис. 5.27.

```
<html>

<head>

<title>Пример использования блоков с абсолютным позиционированием </title>

<style type="text/css"> p {

position:absolute; top:60;

left:100;

background-color:lightgrey; border:double black; margin-left: 10 px;

margin-right: 10 px;

margin-top: 12 px;

padding:50; } img {
```

```

position:absolute; top:140; left:350\ }\

pRE{background-color:white ; border:double black; padding:20; position:absolute;

top:200; left:250

}

</style>

</head>

<body bgcolor=white>

<h3 align=center>пример использования блоков с абсолютным позиционированием</h3> текст текст
текст текст текст текст текст текст текст текст текст текст текст

текст текст текст текст текст текст текст текст текст текст текст текст

<p>

текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст абзаца текст
абзаца текст абзаца текст абзаца

</p>



<pRE>

отформатированный текст

отформатированный текст отформатированный текст отформатированный текст </pRE>

</body>

</html>

```

Допускается указание отрицательных значений для свойств top, bottom, right, left. В этом случае блок будет выходить за рамки своего контейнера (рис. 5.28).



рис. 5.28. Схематичная демонстрация абсолютно позиционируемого блока, выходящего за пределы своего контейнера

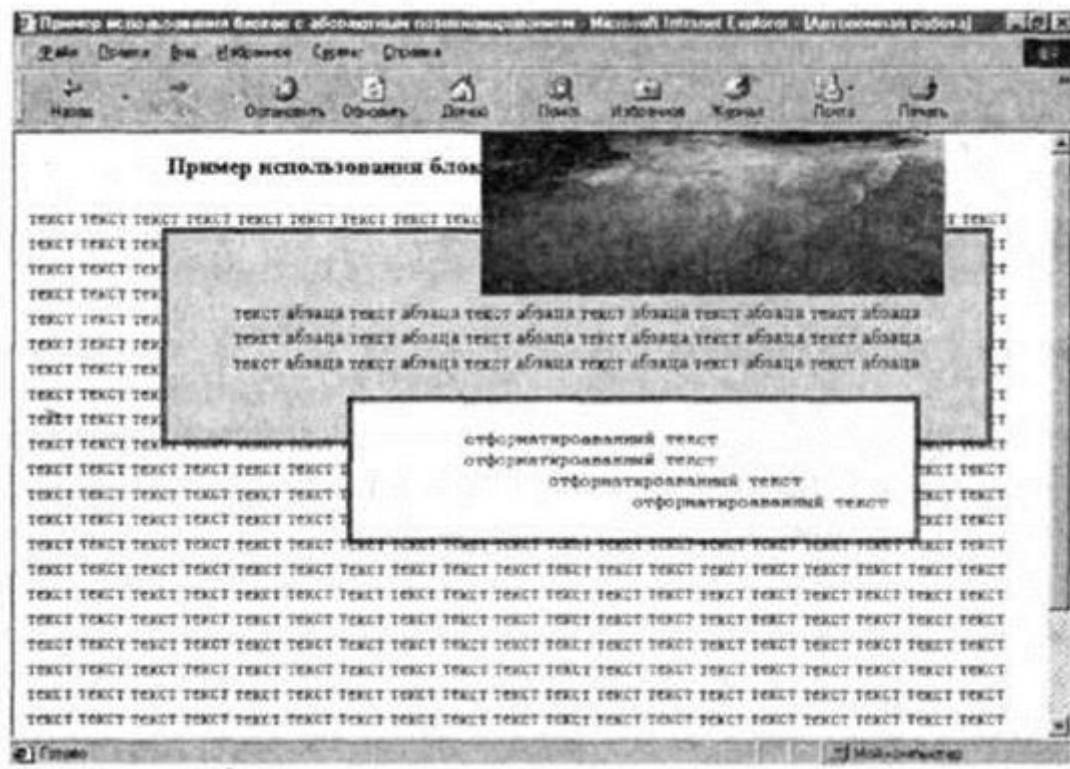


рис. 5.29. Пример документа, в котором для одного из блоков указаны отрицательные координаты относительно начального контейнера

Если отрицательные координаты заданы относительно начального контейнера, то это приводит к утрате той части позиционируемого блока, которая выходит за пределы этого контейнера. Например, если в вышеприведенном примере для изображения указать координату `top:-140`, то часть изображения показана не будет (рис. 5.29).

## Относительное позиционирование

При относительном позиционировании блок сначала размещается на Web-странице в соответствии с нормальным потоком. Затем он может быть смещен относительно своего исходного положения в нормальном потоке. При этом все остальные сестринские структурные блоки располагаются на странице так, как будто относительно позиционированный блок не был перемещен. Т.е. на месте последнего в нормальном потоке отображается пустое пространство, соответствующее его геометрическим размерам.



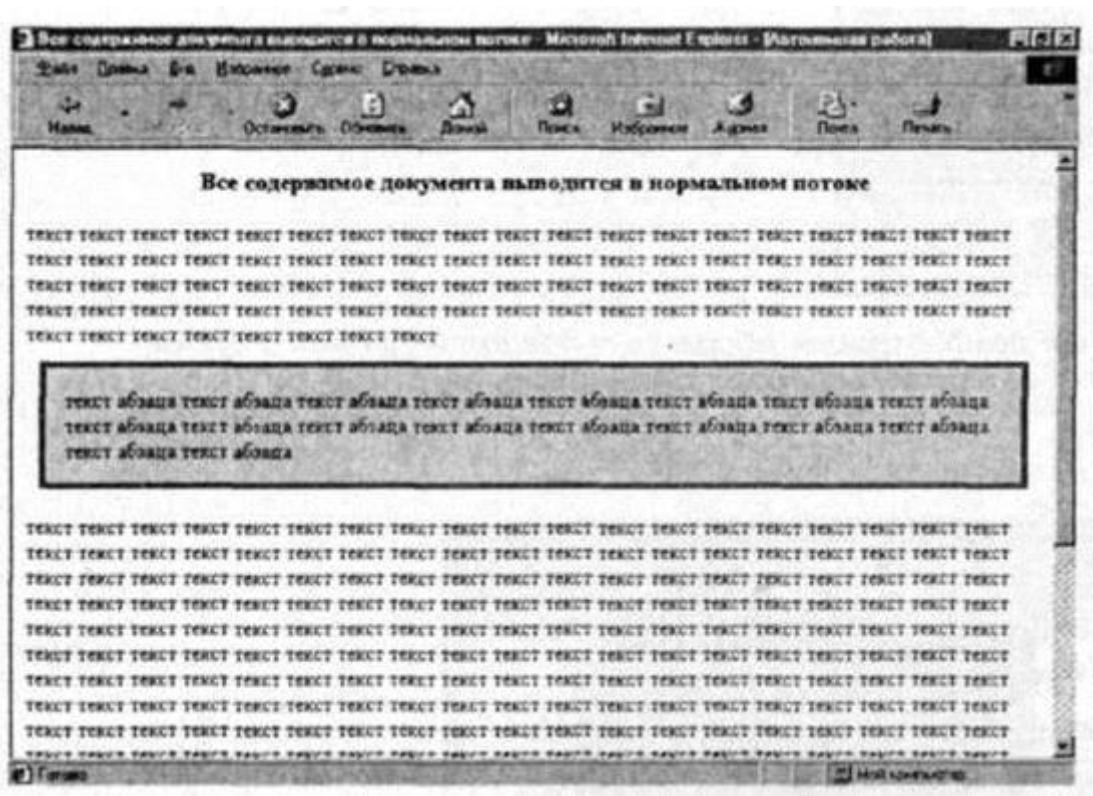


рис. 5.30.\ Абзац ВЫВОДИТСЯ В нормальном потоке

html-элемент порождает относительно позиционированный блок, если для его свойства `position` указано значение `relative`. Относительно позиционируемые блоки, так же как и абсолютно позиционируемые, перекрывают любые блоки нормального потока, независимо от их порядка задания в html-коде документа (до или после относительно позиционируемого блока). Блоки, для которых применена относительная схема позиционирования, могут быть перекрыты только такими же блоками (вместе с их потомками), заданными позднее, или абсолютно позиционируемыми блоками (вместе с их потомками), также заданными позднее.

Потомки относительно позиционируемого блока смещаются вместе с ним, так как вместе с родительским блоком смещается их контейнер. Смещение блока относительно его нормального положения задается с помощью свойств `top` и `left`, которые характеризуют вертикальную и горизонтальную составляющие смещения соответственно.

Примечание.\ Свойства `right` и `bottom` обычно не применяются по причинам, описанным выше.

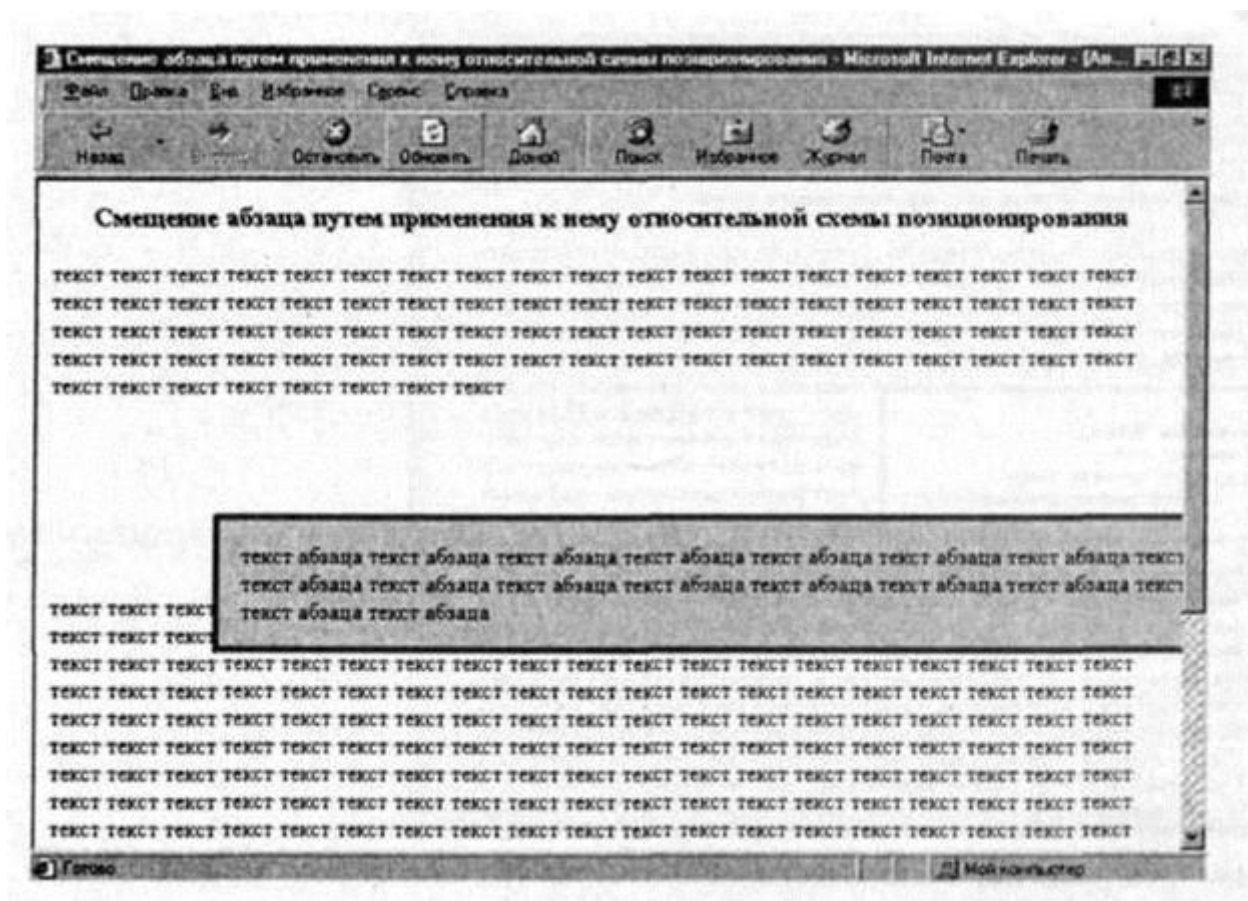


рис. 5.31. Смещение абзаца путем применения к нему относительной схемы позиционирования

Задание значений свойствам `top` и `left` для относительно позиционированного блока аналогично заданию этих свойств для абсолютно позиционируемых блоков. Допускается использование отрицательных значений (в этом случае блок выходит за рамки своего контейнера).

## Перемещаемые блоки. Свойство `float`

Абсолютно позиционируемый блок располагается поверх блоков нормального потока, а также поверх ранее от позиционированных блоков.

Обычно это воспринимается как их положительное качество. Однако могут быть случаи, когда мало просто расположить определенным образом какой-либо блок на Web-странице, а надо, чтобы и окружающие блоки учли его положение и разместились таким образом, чтобы не перекрывать друг друга. Средства языка CSS2 позволяют реализовать некоторые возможности в данном направлении, указав блок как перемещаемый.

Блок является перемещаемым, если для него определено свойство `float`. С помощью этого свойства можно сместить блок либо к левой, либо к правой границе его контейнера. Если, например, блок задан как перемещаемый влево (`float: left`), то он смещается к левой границе контейнера, а все блоки, которые до этого находились левее его, теперь будут располагаться правее. Другими словами, нормальный поток будет обтекать его справа.



рис. 5.32. Пример блока, перемещенного влево

Близкой аналогией свойства float является прикрепление таблиц и изображений к краям документа с помощью задания атрибута align, благодаря чему достигается обтекание их текстом.

Очень удобно в качестве перемещаемого объекта задавать буквицы для абзацев. В соответствие со своим назначением свойство float может принимать следующие значения: left\ — структурный блок задается как перемещаемый влево;

right\ — структурный блок задается как перемещаемый вправо;

none\ — структурный блок никуда не перемещается.



рис. 5.33. Задание буквицы, как перемещенного влево блока

Свойство float можно использовать для относительного позиционирования структурных блоков. При этом в качестве перемещаемого блока будет выступать то исходное положение, относительно которого задается смещение блока, позиционируемого по относительной схеме. Сам этот блок в соответствии со своей схемой, при смещении перестает влиять на положение

окружающих сестринских блоков и перекрывает их, как это было описано выше. Свойство float, описанное для абсолютно позиционируемого блока, игнорируется.

## Многослойный вывод. Свойство Z-index

В предыдущих разделах особое внимание было уделено перекрываемости структурных блоков при их отображении. В общем случае для языка CSS2 каждый структурный блок располагается в трехмерном пространстве. При этом помимо горизонтальных и вертикальных координат, характеризующих положение в плоскости, блоки имеют координату вдоль оси Z, располагаясь один поверх другого. Указание Z-координаты позволяет разработчику явно указывать, какой структурный блок должен располагаться поверх другого. В языке CSS2 эта возможность реализуется с помощью свойства Z-index. Значением этого свойства является целое число, показывающее позиционный уровень блока. Структурный блок с более высоким позиционным уровнем перекрывает блок с более низким позиционным уровнем. Допускается использование отрицательных значений. Каждый родительский блок назначает позиционный контекст своим потомкам, на котором они отображаются.

Пример использования:

```
<html>

<head>

<title>
Демонстрация использования CSS-свойства Z-index </title>

<style type ="text/css"> p ( margin:20; padding:20;
border:solid gray; position:absolute; top:120;
left:200
}

img.first {position: absolute; top:60;
left:100;
Z-index:200}

img.second {position: absolute; top:60;
left:100;
Z-index:2}

body {background-color:lightgrey; Z-index:1000}

</style>

<body>

.. задание элементов тела..... html-документа

.. задание элементов тела..... html-документа

<img id="first" src= >

.. задание элементов телаhtml-документа

<p>...текст абзаца </p>

.. задание элементов тела..... html-документа

<img id="second" src= >

.. задание элементов тела..... html-документа

.. задание элементов телаhtml-документа

</body> </html>
```

При выводе этого документа на экран монитора будет соблюдаться очередность отображения, представленная на рис. 5.34.

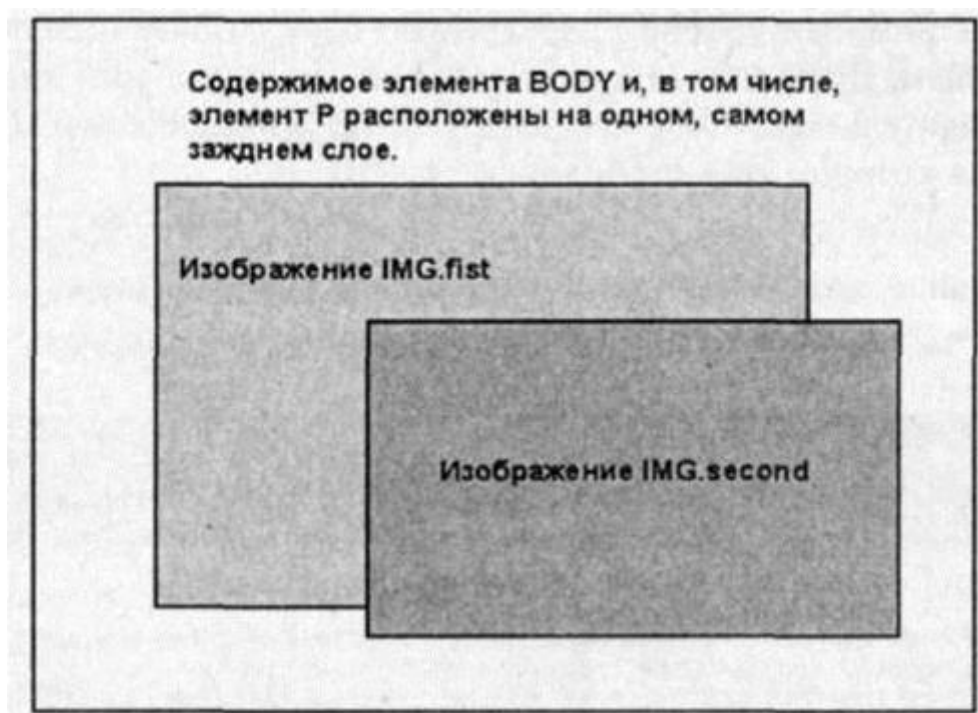


рис. 5.34. Демонстрация многослойной структуры документа

Внимательно просмотрев код документа и порядок его слоев, можно заметить, что, хотя элементу body был присвоен самый большой Z-index, его содержимое отображается на заднем плане. Объясняется это тем, что элемент body является родительским элементом по отношению ко всем другим элементам. Поэтому он задает позиционный контекст им всем. Это означает, что Z-индексы всех дочерних элементов берутся относительно Z-индекса родителя, поэтому ясно, что задание свойства Z-index для элемента body не влечет никаких последствий, то есть бессмысленно.

В качестве значения свойству Z-index можно указывать ключевое слово auto. При этом любой структурный блок с числовым значением Z-index всегда будет перекрывать блок,

значением свойства Z-index которого является ключевое слово auto. Для всех элементов, для которых не указано свойство Z-index, по умолчанию используется значение auto. В вышеприведенном примере демонстрацией этого служит элемент p, для которого не определено свойство Z-index.