

**ЧАСТЬ I**

# **Курс молодого бойца**



# Управление блоком

## Установка размеров

В основе любого элемента разметки (HTML-тега) лежит понятие *блока* — прямоугольной области, выделяемой под элемент разметки. Возможно, вы уже встречались с изображениями, содержащими прозрачные фрагменты, при этом видимые границы изображения фактически могут иметь любую форму. Тем не менее, под любое изображение (как и под все остальные элементы разметки) выделяется прямоугольная область. То же самое касается эффекта скругления углов, которое обычно достигается использованием угловых изображений. Аналогичного эффекта можно будет достигнуть при использовании CSS-свойства `border-radius` и его производных, *когда их поддержка будет реализована браузерами*.

Установка размеров элементов разметки осуществляется с помощью двух CSS-свойств:

- 1) `width` — установка горизонтальных размеров;
- 2) `height` — установка вертикальных размеров.

Значения этих свойств обычно указываются в пикселях или в процентах.

В листинге 1.1 приведен пример установки размеров блока (элемента разметки) с помощью CSS-свойств `width` и `height`.

**Внимание!** Если бы в примере для указания размеров прямоугольников использовались проценты, то их размер вычислялся бы относительно размеров окна браузера.

### Листинг 1.1. Установка размеров блока

```
<!-- <CD>\Samouchitel\block_size.html -->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
    "http://www.w3.org/TR/REC-html40/strict.dtd">

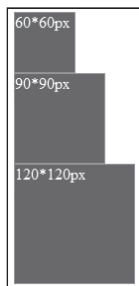
<HTML>
  <HEAD>
    <TITLE>Size</TITLE>
```

```

<STYLE type="text/css">
<!--
    DIV {
        color: white;
        background-color: blue;
        border: 1px solid gray;
    }
-->
</STYLE>
</HEAD>
<BODY>
    <DIV style="width: 60px; height: 60px;">60*60px</DIV>
    <DIV style="width: 90px; height: 90px;">90*90px</DIV>
    <DIV style="width: 120px; height: 120px;">120*120px</DIV>
</BODY>
</HTML>

```

Результат запуска приведенного выше HTML-документа показан на рис. 1.1.



*Рис. 1.1. Результат использования CSS-свойств width и height для установки размеров элементов разметки*

## Установка цвета фона

Установка цвета фона элемента разметки фактически задается либо прямой установкой цвета фона, либо фоновым изображением, причем использование фонового изображения имеет приоритет. Это означает, что если одновременно будет задан цвет фона и фоновое изображение, то пользователь увидит “заливку” фона фоновым изображением. Это сделано в расчете на ситуацию, когда фоновое изображение не может быть подгружено. Поэтому часто при установке фоновых изображений также устанавливается цвет фона, близкий к цветовой гамме фонового изображения (на тот случай, если изображение не найдено по указанному сетевому пути или пользователь использует режим посещения сайтов без автоматической загрузки изображений).

Установка цвета фона осуществляется с помощью одного из двух CSS-свойств:

- 1) `background-color` — установка цвета фона;
- 2) `background` — установка любого из параметров фона, в том числе и цвета фона (фактически повторяет использование свойства `background-color`, при этом позволяет одновременно устанавливать другие параметры фона).

Значение цвета фона устанавливается:

- 1) в формате RGB (в полной или сокращенной форме);
- 2) с помощью predefined констант цвета (см. приложение Г, “Стандартные цвета”).

Установка фонового изображения рассмотрена в главе 5, “Управление изображением”.

В листинге 1.2 приведен пример установки цвета фона с помощью CSS-свойств `background` и `background-color`.

### Листинг 1.2. Установка цвета фона

```
<!-- <CD>\Samouchitel\background_color.html -->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
           "http://www.w3.org/TR/REC-html40/strict.dtd">

<HTML>
  <HEAD>
    <TITLE>Size</TITLE>
    <STYLE type="text/css">
      <!--
        DIV {
          width: 60px; height: 60px;
          border: 1px solid gray;
        }
      -->
    </STYLE>
  </HEAD>
  <BODY>
    <DIV style="background: blue;"></DIV>
    <DIV style="background-color: green;"></DIV>
  </BODY>
</HTML>
```

Нет смысла приводить результат запуска приведенного выше HTML-документа (цветное изображение) в черно-белой книге.

## Тип блока

Тип блока является одним из самых тонких понятий языка HTML и каскадных таблиц стилей CSS. Отсюда идет различие двух блочных элементов разметки (тегов) `<DIV>` и `<SPAN>`. Кстати, этот вопрос является классическим при собеседовании с Web-программистами, дизайнерами и верстальщиками Web-страниц. Понимание этого различия дает реальную оценку уровня кандидата и, как следствие, его способности к верстке Web-страниц.

Тег `<DIV>` в стандартном состоянии занимает всю ширину родительского элемента разметки. При этом после перехода на новую строку происходит отображение тега `<DIV>` и его содержимого, а следующий элемент разметки занимает следующую строку в родительском элементе. В принципе, аналогично ведет себя тег `<P>`, но он

предназначен в первую очередь для выделения текстовых фрагментов (в простейшем варианте абзацев).

Тег `<SPAN>` в стандартном состоянии занимает площадь (длину строки родительского элемента разметки), необходимую для отображения содержимого этого тега.

Оба тега `<DIV>` и `<SPAN>` в основном используются для создания каркаса Web-страницы.

Управление типом блока можно осуществлять с помощью CSS-свойства `display`. Наиболее интересны следующие значения-константы этого свойства:

- 1) `block` — создает переход на новую строку сверху и снизу от блока, блок занимает всю ширину родительского элемента (аналогично свойствам по умолчанию тегов `<P>` и `<DIV>`);
- 2) `inline` — блок занимает только необходимое для отображения содержимого место в родительском элементе (аналогично свойствам по умолчанию тега `<SPAN>`).

В листинге 1.3 приведен пример установки типа блока для тегов `<DIV>` и `<SPAN>` при помощи CSS-свойства `display`.

### Листинг 1.3. Установка цвета фона

```
<!-- <CD>\Samouchitel\block_type.html -->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
    "http://www.w3.org/TR/REC-html40/strict.dtd">
<HTML>
  <HEAD>
    <TITLE>Block type</TITLE>
    <STYLE type="text/css">
      <!--
        DIV, SPAN { border: 2px solid black; }
      -->
    </STYLE>
  </HEAD>
  <BODY>
    <DIV>DIV #1</DIV>
    <DIV>DIV #2</DIV>
    <DIV>DIV #3</DIV><BR>

    <SPAN>SPAN #1</SPAN>
    <SPAN>SPAN #2</SPAN>
    <SPAN>SPAN #3</SPAN><BR><BR>

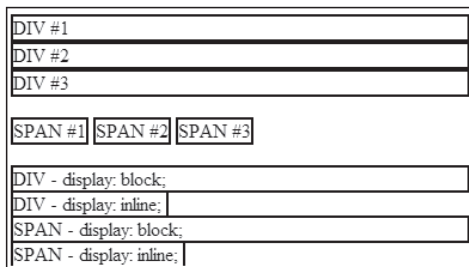
    <DIV style="display: block;">
      DIV - display: block;
    </DIV>
    <DIV style="display: inline;">
      DIV - display: inline;
    </DIV>
    <SPAN style="display: block;">
      SPAN - display: block;
    </SPAN>
    <SPAN style="display: inline;">
```

```

        SPAN - display: inline;
    </SPAN>
</BODY>
</HTML>

```

Результат запуска приведенного выше HTML-документа показан на рис. 1.2.



*Рис. 1.2. Результат использования CSS-свойства display для тегов <DIV> и <SPAN> со значениями block и inline*

### Вопрос для самопроверки

Обычно результат размещения тега <SPAN> после тега <DIV> очевиден. А что будет, если поместить тег <DIV> после тега <SPAN>?

## Внешние отступы

Установка внешних отступов наиболее актуальна для изображений при внедрении их внутрь текста, а также при размещении текстового контента (текста) внутри каркаса Web-страницы при “резиновом” дизайне (в простейшем случае ширина Web-страницы и ее элементов подстраивается под изменяющиеся горизонтальные размеры окна браузера).

Установка внешних отступов элемента разметки осуществляется с помощью CSS-свойства margin и его производных для верхнего, правого, нижнего и левого краев (margin-top, margin-right, margin-bottom, margin-left). Обычно значение этих свойств устанавливается в пикселях. Подробнее об использовании CSS-свойства margin и его производных см. в главе 13, “Свойства рамки и отступов для блока”.

В листинге 1.4 приведен пример установки внешних отступов с помощью CSS-свойства margin. В примере приведены четыре синих квадрата, которые помещены в белые прямоугольники. Позиции квадратов определяются внешними отступами синих квадратов, устанавливаемыми значениями CSS-свойства margin.

### Листинг 1.4. Пример установки внешних отступов

```

<!-- <CD>\Samouchitel\block_external_standoff.html -->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
        "http://www.w3.org/TR/REC-html40/strict.dtd">
<HTML>
  <HEAD>
    <TITLE>Standoff margin</TITLE>

```

```

<STYLE type="text/css">
<!--
  DIV {
    background-color: blue;
    border: 1px solid gray;
    width: 30px; height: 30px;
  }
  TABLE {
    border: 1px solid gray;
  }
-->
</STYLE>
</HEAD>
<BODY>
  <DIV style="margin: 0px;"></DIV>
  <TABLE><TR><TD><DIV style="margin: 30px;"></DIV></TD></TR></
TABLE>
  <TABLE><TR><TD><DIV style="margin: 90px;"></DIV></TD></TR></
TABLE>
  <TABLE><TR><TD><DIV style="margin: 10px 30px 60px 90px;">
  </TD></TR></TABLE>
</BODY>
</HTML>

```

Результат запуска приведенного выше HTML-документа показан на рис. 1.3.

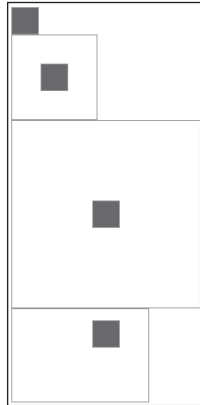


Рис. 1.3. Пример использования CSS-свойства margin

## Внутренние отступы

Установка внутренних отступов наиболее актуальна при формировании каркаса Web-страницы как при “табличной” верстке (на основе тега <TABLE>), так и при “дивной” (на основе тега <DIV>).

Установка внутренних отступов элемента разметки осуществляется с помощью CSS-свойства padding и его производных для верхнего, правого, нижнего и левого краев (padding-top, padding-right, padding-bottom и padding-left). Обычно



значение этих свойств устанавливается в пикселях. Подробнее об использовании CSS-свойства padding и его производных см. в главе 13, “Свойства рамки и отступов”.

В листинге 1.5 приведен пример установки внешних отступов с помощью CSS-свойства padding. В примере создаются четыре синих прямоугольника, в которые помещены красные квадраты. Позиции квадратов определяются внутренними отступами синих прямоугольников, устанавливаемыми значениями CSS-свойства padding.

#### Листинг 1.5. Пример установки внутренних отступов

```
<!-- <CD>\Samouchitel\block_internal_standoff.html -->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
          "http://www.w3.org/TR/REC-html40/strict.dtd">
<HTML>
  <HEAD>
    <TITLE>Standoff padding</TITLE>
    <STYLE type="text/css">
      <!--
        DIV {
          background-color: blue;
          border: 1px solid gray;
          width: 30px; height: 30px;
        }
      -->
    </STYLE>
  </HEAD>
  <BODY>
    <DIV style="padding: 0px;"><DIV class="red"></DIV></DIV>
    <DIV style="padding: 30px;"><DIV class="red"></DIV></DIV>
    <DIV style="padding: 60px;"><DIV class="red"></DIV></DIV>
    <DIV style="padding: 10px 30px 60px 90px;"><DIV class="red">
    </DIV></DIV>
  </BODY>
</HTML>
```

Результат запуска приведенного выше HTML-документа показан на рис. 1.4.

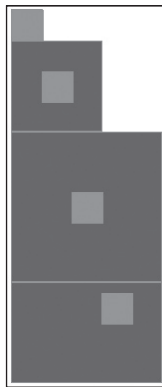


Рис. 1.4. Пример использования CSS-свойства padding

## Установка ширины, цвета и типа рамки

Каждый элемент разметки имеет рамку, которая по умолчанию не отображается. Как правило, с понятием рамки обычно ассоциируется рамка таблицы или ее ячеек. Тем не менее рамка может быть установлена для любого элемента разметки.

Наиболее общими являются следующие CSS-свойства для создания рамки:

- 1) `border-width` — устанавливает ширину рамки;
- 2) `border-style` — устанавливает стиль линии рамки;
- 3) `border-color` — устанавливает цвет рамки;
- 4) `border` — результирующее свойство, позволяющее задавать одновременно ширину, стиль линии и цвет рамки.

Все перечисленные выше CSS-свойства имеют подсвойства для верхней, левой, правой и нижних составляющих. Более подробное описание использования CSS-свойства `border` и его производных см. в главе 13, “Свойства рамки и отступов”.

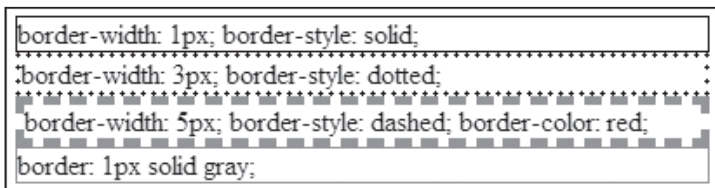
В листинге 1.6 приведен пример создания рамки с помощью CSS-свойств `border-width`, `border-style`, `border-color` и `border`. В примере создаются четыре прямоугольника, к которым применяются перечисленные выше CSS-свойства в разных комбинациях.

### Листинг 1.6. Пример создания рамки на основе `border`-свойств

```
<!-- <CD>\Samouchitel\block_border1.html -->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
    "http://www.w3.org/TR/REC-html40/strict.dtd">
<HTML>
  <HEAD>
    <TITLE>Border</TITLE>
  </HEAD>
  <BODY>
    <DIV style="border-width: 1px; border-style: solid;">
      border-width: 1px; border-style: solid;
    </DIV>
    <DIV style="border-width: 3px; border-style: dotted;">
      border-width: 3px; border-style: dotted;
    </DIV>
    <DIV style="border-width: 5px; border-style: dashed; border-color:
    ↵ red;">
      border-width: 5px; border-style: dashed; border-color: red;
    </DIV>
    <DIV style="border: 1px solid gray;">
      border: 1px solid gray;
    </DIV>
  </BODY>
</HTML>
```

---

Результат запуска приведенного выше HTML-документа показан на рис. 1.5.



**Рис. 1.5.** Результат использования CSS-свойств `border-width`, `border-style`, `border-color` и `border`

Часто на практике необходимо создать только одну из составляющих рамки. CSS предлагает набор соответствующих свойств, которые образуются включением слов `top`, `left`, `right` и `bottom`. Ниже приведены маски для свойств, применяемые к элементам, составляющим рамку:

```
border-*-width
border-*-style
border-*-color
border-*,
```

где вместо звездочки помещается одно из слов (`top`, `left`, `right` и `bottom`), означающее верхнюю, левую, правую или нижнюю составляющую рамки соответственно.

В листинге 1.7 приведен пример создания рамки с помощью производных от CSS-свойств `border-width`, `border-style`, `border-color` и `border`. В примере создаются четыре прямоугольника, к которым применяются перечисленные выше CSS-свойства в разных комбинациях.

#### Листинг 1.7. Пример создания рамки на основе производных от border-свойств

```
<!-- <CD>\Samouchitel\block_border2.html -->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
    "http://www.w3.org/TR/REC-html40/strict.dtd">
<HTML>
  <HEAD>
    <TITLE>Border</TITLE>
    <STYLE type="text/css">
      <!--
        DIV
        {
          border-width: 1px;
          border-style: solid;
          height: 50px;
          margin-bottom: 5px;
        }
      //-->
    </STYLE>
  </HEAD>
  <BODY>
    <DIV style="border-right-width: 10px;"
      border-right-width: 10px;
    </DIV>
    <DIV style="border-left-width: 5px; border-left-style:
      ↵ dotted;">
```

```

border-left-width: 5px; border-left-style: dotted;
</DIV>
<DIV style="border-bottom-style: dashed; border-color: red;">
border-bottom-style: dashed; border-color: red;
</DIV>
<DIV style="border-bottom: 1px dashed blue;">
border-bottom: 1px dashed blue;
</DIV>
</BODY>
</HTML>

```

Результат запуска приведенного выше HTML-документа показан на рис. 1.6.

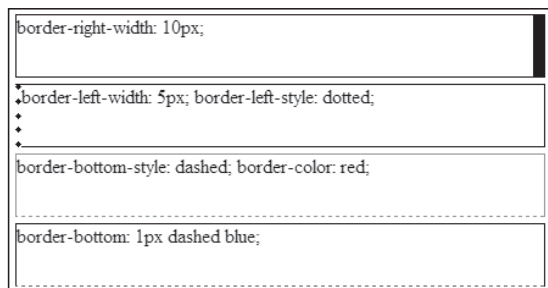


Рис. 1.6. Результат использования производных от CSS-свойств `border-width`, `border-style`, `border-color` и `border`

## Создание якорей

Якоря создаются в системе абсолютных координат, при этом за точку отсчета берется левый верхний угол окна браузера. Якоря позволяют закрепить какие-либо объекты (элементы графики, изображения или элементы пользовательского интерфейса) на фиксированном расстоянии от границ окна браузера.

Якоря создаются на основе следующих CSS-свойств:

- 1) `top` — устанавливает расстояние от верхней границы браузера;
- 2) `left` — устанавливает расстояние от левой границы браузера;
- 3) `right` — устанавливает расстояние от правой границы браузера;
- 4) `bottom` — устанавливает расстояние от нижней границы браузера.

В листинге 1.8 приведен пример создания якорей с помощью CSS-свойств `top`, `left`, `right` и `bottom`. В примере создается прямоугольник, размеры которого изменяются в зависимости от размеров окна браузера, причем расстояние от любой стороны прямоугольника и окна браузера равно ста пикселям.

### Листинг 1.8. Создание якорей с привязкой ко всем сторонам браузера

```

<!-- <CD>\Samouchitel\block_anchors1.html -->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
"http://www.w3.org/TR/REC-html40/strict.dtd">
<HTML>

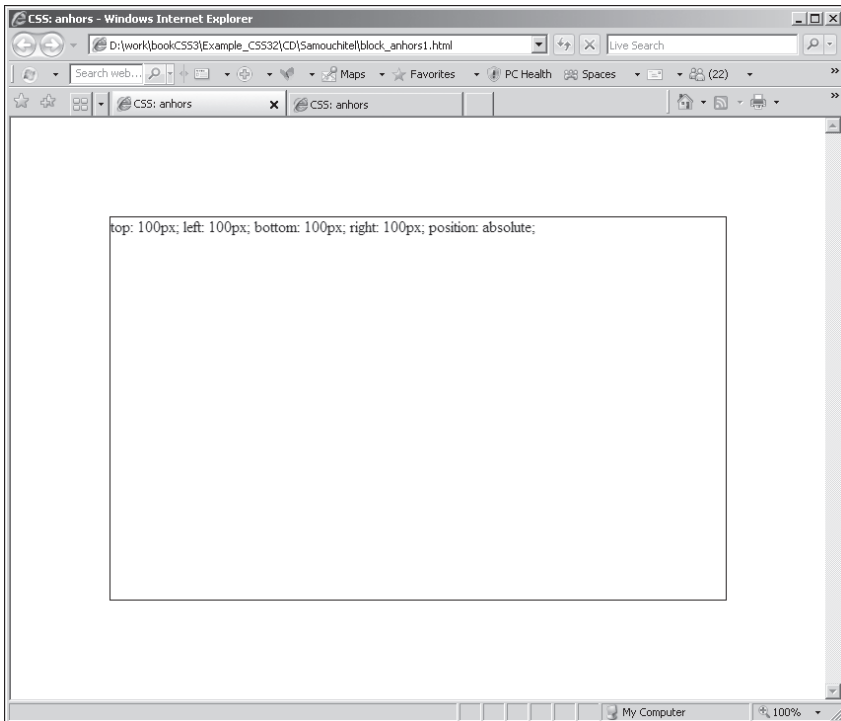
```

```

<HEAD>
  <TITLE>Anhors</TITLE>
  <STYLE type="text/css">
  <!--
    SPAN
    {
      border: 1px solid black;
    }
  -->
</STYLE>
</HEAD>
<BODY>
  <SPAN style="top: 100px; left: 100px; bottom: 100px; right:
100px; position: absolute;">
    top: 100px; left: 100px; bottom: 100px; right: 100px;
  </SPAN>
  position: absolute;
</BODY>
</HTML>

```

Результат запуска приведенного выше HTML-документа показан на рис. 1.7.



**Рис. 1.7.** Создание якорей с привязкой ко всем сторонам браузера с помощью CSS-свойств `top`, `left`, `right` и `bottom`

Аналогичным образом может осуществляться привязка к одной, двум и трем сторонам браузера. В листинге 1.9 приведен пример создания якорей с привязкой к двум сторонам окна браузера. Это особенно удобно для всевозможных вспомогательных окон в DHTML-(AJAX-) приложениях. В примере создаются четыре прямоугольника, которые привязываются к углам окна браузера (к двум сторонам браузера).

### Листинг 1.9. Создание якорей с привязкой к углам окна браузера

```
<!-- <CD>\Samouchitel\block_anhors2.html -->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
    "http://www.w3.org/TR/REC-html40/strict.dtd">
<HTML>
  <HEAD>
    <TITLE>Anhors</TITLE>
    <STYLE type="text/css">
      <!--
        SPAN
        {
          border: 1px solid black;
        }
      -->
    </STYLE>
  </HEAD>
  <BODY>
    <SPAN style="top: 100px; left: 100px; position: absolute;">
      top: 100px; left: 100px; position: absolute;
    </SPAN>

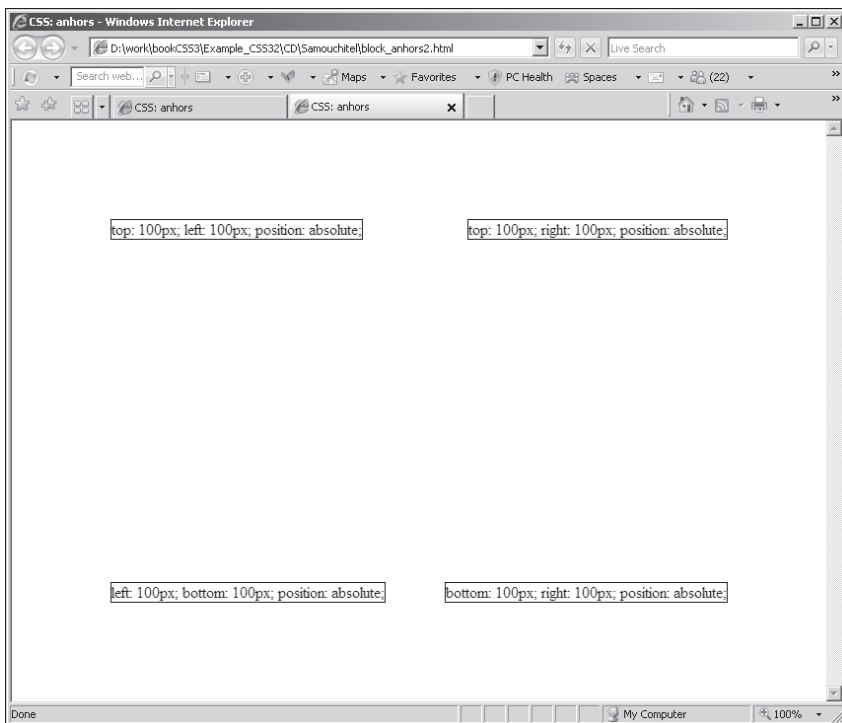
    <SPAN style="top: 100px; right: 100px; position: absolute;">
      top: 100px; right: 100px; position: absolute;
    </SPAN>

    <SPAN style="left: 100px; bottom: 100px; position: absolute;">
      left: 100px; bottom: 100px; position: absolute;
    </SPAN>

    <SPAN style="bottom: 100px; right: 100px; position: absolute;">
      bottom: 100px; right: 100px; position: absolute;
    </SPAN>
  </BODY>
</HTML>
```

---

Результат запуска приведенного выше HTML-документа показан на рис. 1.8.



*Рис. 1.8. Создание якорей с привязкой к углам окна браузера с помощью CSS-свойств top, left, right и bottom*

## Абсолютное позиционирование

Абсолютное позиционирование достигается с помощью тех же CSS-свойств top, left, которые используются для создания якорей с привязкой к верхней и левой границам окна браузера. Абсолютную позицию можно сравнить с якорем привязки к началу отсчета координат (т.е. к левому верхнему углу окна браузера).

В листинге 1.10 приведен пример абсолютного позиционирования с помощью CSS-свойств top и left. В примере создаются четыре прямоугольника, которые устанавливаются в различные позиции окна браузера.

### Листинг 1.10. Абсолютное позиционирование

```
<!-- <CD>\Samouchitel\block_absolute_position.html -->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
    "http://www.w3.org/TR/REC-html40/strict.dtd">
<HTML>
  <HEAD>
    <TITLE>Absolute position</TITLE>
    <STYLE type="text/css">
      <!--
        SPAN
        {
```

```

        border: 1px solid black;
        background-color: silver;
    }
-->
</STYLE>
</HEAD>
<BODY>
    <SPAN style="left: 100px; top: 100px; position: absolute;">
        100, 100
    </SPAN>
    <SPAN style="left: 300px; top: 200px; position: absolute;">
        300, 200
    </SPAN>
    <SPAN style="left: 150px; top: 180px; position: absolute;">
        150, 180
    </SPAN>
    <SPAN style="left: 70px; top: 250px; position: absolute;">
        70, 250
    </SPAN>
</BODY>
</HTML>

```

Результат запуска приведенного выше HTML-документа показан на рис. 1.9.

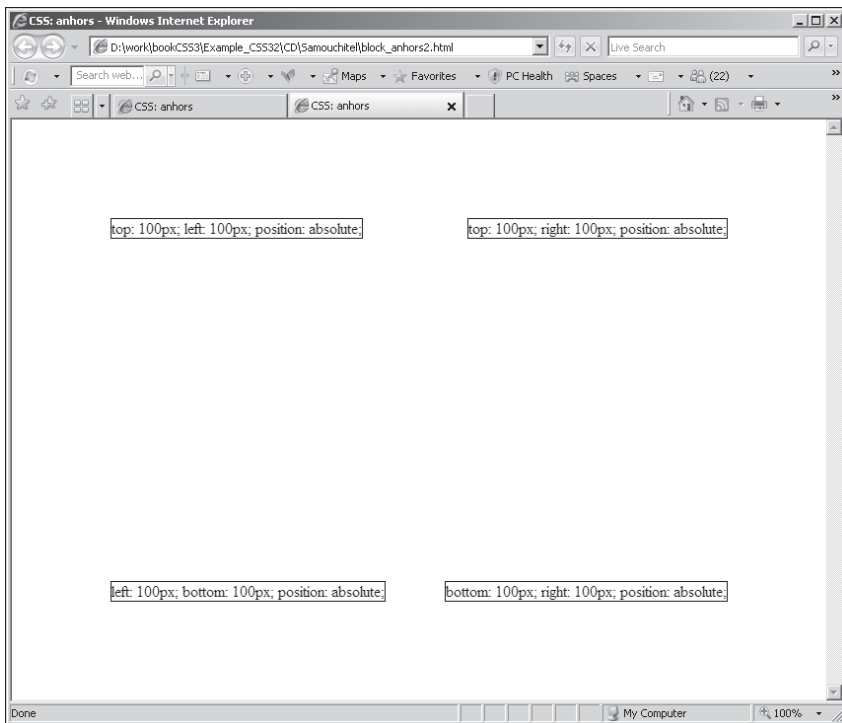


Рис. 1.9. Абсолютное позиционирование с помощью CSS-свойств *top* и *left*