**Лабораторная работа № 5**

**Понятие блочной модели CSS**

**Цель работы:** познакомиться с блочной модель, изучить свойства блочных элементов.

**Теоретические сведения для выполнения работы**

**Блочные элементы CSS**

В CSS существуют два различных типа элементов: блочные и строчные. Строчные элементы не создают отступы до и после, они отображаются на одной строке с содержимым рядом стоящих элементов. Примерами строчных элементов являются strong, em, img, a, span.

В блочных элементах создается разрыв строки перед элементом и после него. Например, абзац ***p*** создает блок, отделенный от элементов, расположенных выше и ниже его. Другими примерами являются заголовки, контейнеры div, таблицы, списки и элементы списков.

**Свойства блочных элементов**

Браузер обрабатывает все элементы как небольшие блоки. Основной частью каждого блока элемента является область содержимого. Область содержимого окружена произвольным количеством следующих свойств (рис. 5.1):

1. *padding* — отступ, пространство между контентом и границей;

2. *border* — граница, линия вдоль каждого края блока;

3. *margin* — поле, отделяет один элемент от другого.

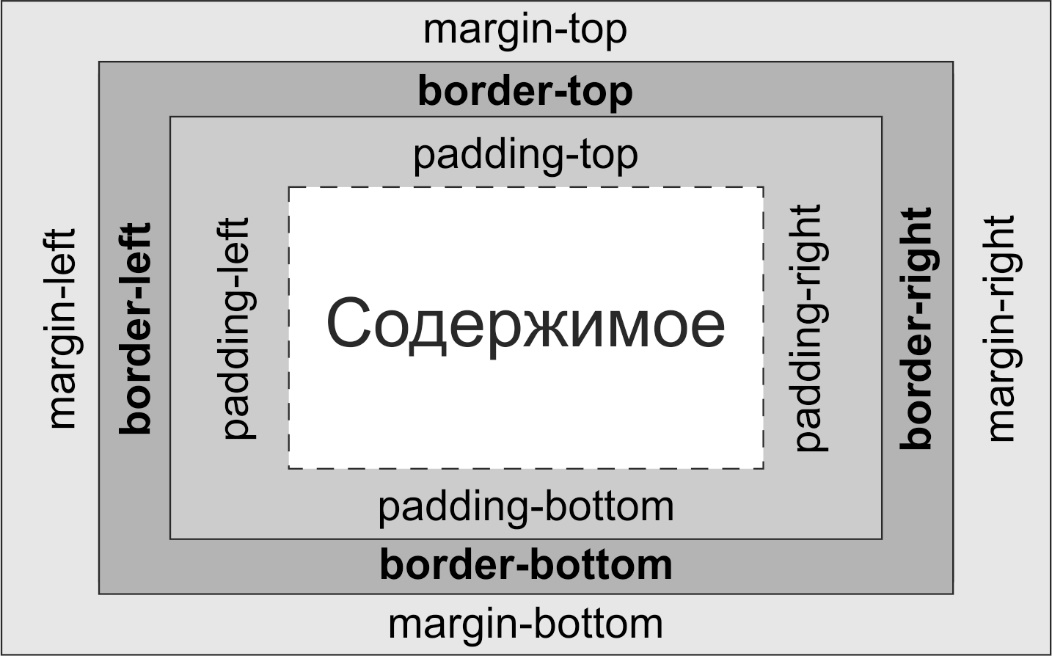


Рис. 5.1

Для форматирования элемента можно использовать любые из этих свойств в любом сочетании или все сразу. Можно использовать любые единицы измерения, принятые в языке CSS, для определения размеров полей и отступов, например:

margin-right: 20px;

padding-top: 3em;

margin-left: 10%

Можно использовать сокращенные варианты свойств *margin* и *padding* для быстрой установки всех четырех параметров одновременно. Они должны указываться в следующей последовательности: сверху, справа, снизу и cлева.

margin: 0 10px 10px 20px;

padding: 10px 5px 5px 10px;

Также блоковые элементы могут быть выровнены по центру установкой *margin* с левой и правой стороны значения ***auto***.

**Свойство display**

В некоторых случаях требуется, чтобы строчные элементы вели себя так же, как блочные, или наоборот. В языке CSS позволяет это сделать свойство *display*. С его помощью можно заставить блочный элемент функционировать как строчный: ***display*: *inline***. И наоборот, можно сделать так, чтобы строчные элементы, например изображение или ссылка, вели себя как блочные: ***display*: *block***.

Также можно заставить элемент действовать и как блочный, и как строчный. Значение ***inline-block*** не создает разрывов ни до, ни после элемента и одновременно заставляет элемент подчиняться верхним и нижним полям и отступам, а также настройкам высоты: ***display: inline-block***.

**Свойства границ**

Можно управлять тремя различными свойствами любой из границ: *color* (цвет), *width* (ширина) и *style* (стиль). Для ширины границы width используются любые единицы измерения каскадных таблиц стилей (кроме процентов) или ключевые слова ***thin*** (тонкая линия), ***medium*** (средняя) и ***thick*** (толстая). Самые распространенные единицы измерения для данного свойства — пикселы.

Свойство *style* управляет типом линии границы. В каскадных таблицах стилей для границ имеются следующие стили: ***solid, dotted, dashed, double, groove, ridge, inset, outset, noneи hidden***. Ключевые слова none и hidden работают одинаково: они полностью удаляют границы. Но значение none удобно использовать для удаления границы с одной стороны элемента. Для установки границ можно использовать сокращенную запись или расширенную следующим образом:

border: 2px double #FFCC33;

или

border-width: 2px;

border-style: double;

border-color: #FFCC33;

Следует отметить, что каждая сторона имеет свой набор из трех свойств, которые удобно использовать для отмены одного. Правая граница *border-right-width*, *border-right-style* и *border-right-color*. Левая, верхняя и нижняя границы имеют похожие свойства: border-left-width, border-left-style и т. д. Однако можно задать собственные значения сразу для каждой стороны границы, используя сокращенную запись свойства *border-width*, *border-style* и *border-color*. Например, правило ***border-width: 10px 5px 15px 13px;*** применит четыре различных значения ширины для каждой из сторон (верхней, правой, нижней и левой).

В языке CSS существует свойство *border-radius*, позволяющее дизайнерам добавлять скругления к одному или нескольким углам элемента. Для каждого угла можно предоставить отдельные значения, задав четыре параметра. Объявление свойства имеет следующий вид:

border-radius: 0 30px 10px 5px;

Сначала задается числовое значение для левого верхнего угла блока, а затем по часовой стрелке для всех остальных углов.

Чтобы добавить эллиптические углы, нужно создать следующее объявление:

border-radius: 40px/20px;

**Добавление тени**

Для добавления теней к блоку, обрамляющему элемент, используется свойство *box-shadow*. По сравнению с text-shadow тень можно добавлять внутри блока с помощью ключевого слова inset.

Основной синтаксис свойства *box-shadow* следующий:

box-shadow: inset 4px 4px 8px 12px rgba(0,0,0,.75);

Первое значение задает горизонтальное смещение, то есть это значение приводит к перемещению тени влево или вправо от элемента. Положительное число приводит к перемещению тени вправо, а отрицательное число — влево.

Второе значение задает вертикальное смещение — позицию тени либо над элементом, либо под ним. При положительном значении тень помещается ниже нижнего края блока, а при отрицательном значении тень помещается над верхним краем блока.

Третье значение задает радиус размытия тени. Оно определяет степень размытости и ширины тени. Чем выше значение, тем более размытой и тусклой становится тень.

Последнее значение задает цвет отбрасываемой тени. Можно вос-пользоваться любым обозначением цвета, принятым в языке CSS, но RGBA-значения позволяют управлять прозрачностью тени, делая ее более реалистичной.

В качестве четвертого значения (между радиусом размытия тени и ее цветом) можно также добавить расширение. Это приведет к расширению тени на указанное значение.

**Размеры блочных элементов**

Браузеры вычисляют ширину блочного элемента, складывая значения свойств *border*, *padding* и *width*. Таким образом, необходимо производить математические вычисления для определения занимаемого экранного пространства. На этот случай каскадные таблицы стилей предлагают свойство *box-sizing*, позволяющее изменить порядок вычисления браузером экранной ширины (и высоты) элемента. Свойство *box-sizing* имеет следующие значения:

1. ***content-box*** добавляет ширину границ и толщину отступа к значениям, установленным для свойств ширины и высоты, чтобы определить экранную ширину и высоту заданного элемента.

2. ***padding-box*** сообщает браузеру, что при установке для стиля свойства ширины или высоты должны включать отступы как часть своего значения

3. ***border-box*** сообщает браузеру о необходимости включения в качестве составляющей части значений свойств *width* и *height* толщину как отступа, так и границ. Это значение можно использовать для универсального селектора.

Когда содержимое форматируемого элемента имеет размеры больше определенных свойствами *width* и *height*, то используется свойство *overflow* со следующими значениями:

– ***visible*** имеет тот же эффект, что отсутствие установки свойства;

– ***scroll*** позволяет добавить полосы прокрутки.

– ***auto*** выполняет ту же функцию, что и ***scroll***, но полосы прокрутки в данном случае появляются только при необходимости.

– ***hidden*** скрывает любое содержимое, выходящее за пределы блочного элемента.

**Задания к лабораторной работе № 5**

**Задание 1** Создайте HTML документ с заголовком Блочная модель, в котором будет создано семь блоков div согласно рис. 5.2:

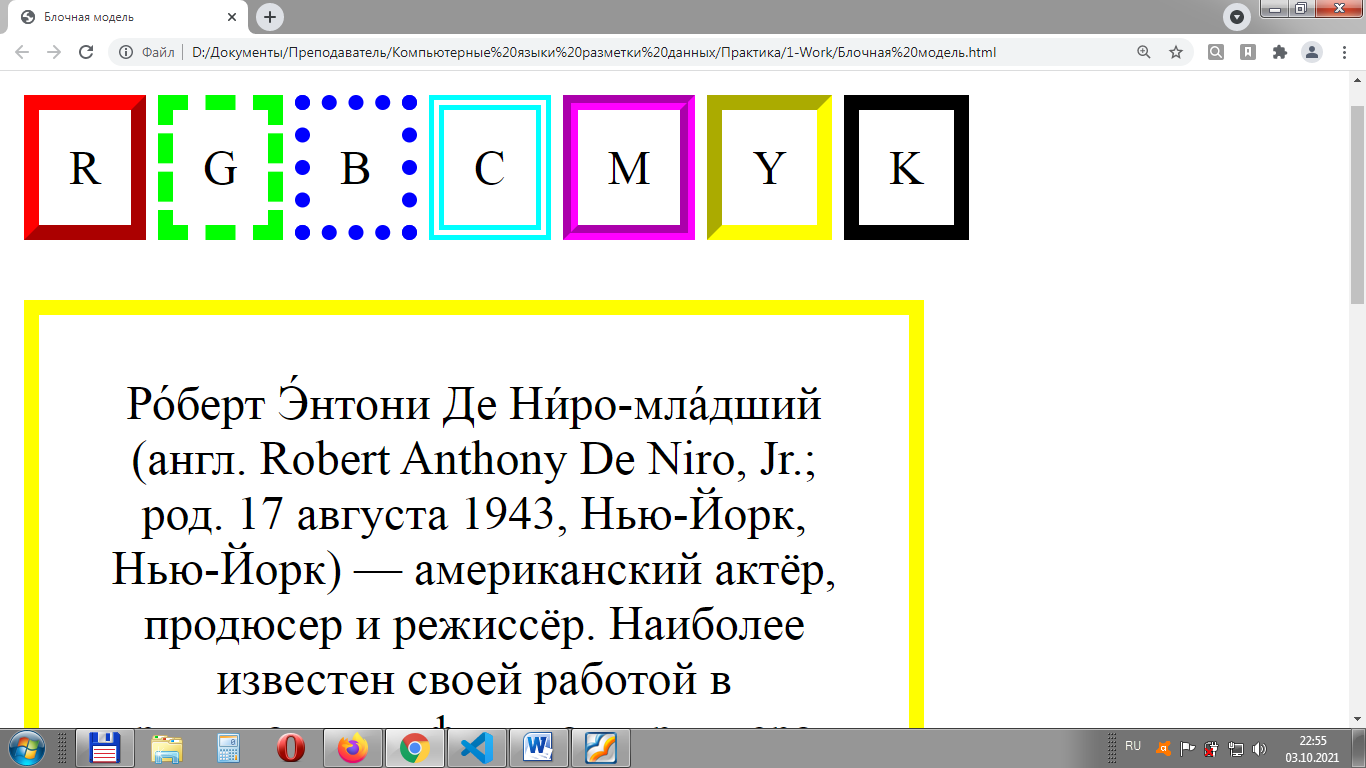


Рис. 5.2

**1.1** все блоки имеют должны располагаться в одну линию;

**1.2** цвета изменяются в следующем порядке: красный, зеленый, синий, голубой, пурпурный, желтый, черный. Задать их с помощью rgb;

**1.3** задать все отступы по 10px, отступ сверху 10px.

**Задание 2** Создайте в этом же документе 3 абзаца, в каждый из абзацев добавьте произвольные строчные элементы (все должны быть разные). Для них задать отступы и поля по 5 px, фон #FFA500, сплошная граница #FF4500. Также для каждого абзаца задать следующие свойства

**2.1** Для первого абзаца задать ширину 400 px и выравнивание по центру, сплошную границу #FF7F50;

**2.2** для второго абзаца установите сплошную границу красного цвета с толщиной 5px с эллиптическими углами округления 20 px/40 px; добавьте внутреннюю тень rgba(0,0,0,.5) цвета со смещением по горизонтали вправо и по вертикали вниз на 2px, с размытием 8px и расширением 8 px. Установите значение auto свойства overflow;

**2.3** третий абзац должен иметь сплошную границу #FF6347.

**Задание 3** Скопируйте block.html из папки labs. Добавьте необходимые элементы, а также свойства абзацам согласно их описанию.

**Контрольные вопросы**

1. Какие элементы относятся к блочным? Приведите примеры

<form>, <h1>,...,<h6>, <hr>, <ol>, <p>, <table>, <ul>

2. Какие элементы относятся к строчным? Приведите примеры

<img>, <span>, <a>

3. Как задать свойства блочных элементов строчным?

***display*: *block***.

4. Как задать свойства строчных элементов блочным?

***display*: *inline***

4. Сокращенная запись свойства *margin*?

margin:(top, right, bottom, left);

5. Какие значения имеет свойство *box-shadow*?

Первое значение задает горизонтальное смещение, то есть это значение приводит к перемещению тени влево или вправо от элемента. Положительное число приводит к перемещению тени вправо, а отрицательное число — влево.

Второе значение задает вертикальное смещение — позицию тени либо над элементом, либо под ним. При положительном значении тень помещается ниже нижнего края блока, а при отрицательном значении тень помещается над верхним краем блока.

Третье значение задает радиус размытия тени. Оно определяет степень размытости и ширины тени. Чем выше значение, тем более размытой и тусклой становится тень.

Последнее значение задает цвет отбрасываемой тени. Можно вос-пользоваться любым обозначением цвета, принятым в языке CSS, но RGBA-значения позволяют управлять прозрачностью тени, делая ее более реалистичной.

В качестве четвертого значения (между радиусом размытия тени и ее цветом) можно также добавить расширение. Это приведет к расширению тени на указанное значение.

6. Как разместить блочный элемент по центру?

margin:auto;

7. Какие свойства имеют блочные элементы?

Для **блочных элементов** характерны следующие особенности. По ширине **блочные элементы** занимают всё допустимое пространство. Если **элемент** располагается непосредственно внутри <body>, то он будет занимать всю ширину веб-страницы. Если один **элемент** находится внутри другого, то он будет занимать всю ширину своего родителя.

8. Какие свойства имеют граница блочных элементов?

: ***solid, dotted, dashed, double, groove, ridge, inset, outset, noneи hidden***.

8. Каким образом определяется ширина блочного элемента?

width: 100px;

9. Для чего используется свойство *box-sizing*?

Применяется для изменения алгоритма расчёта ширины и высоты элемента. Аналогично обстоит и с высотой блока. ... **Свойство box**-**sizing** позволяет изменить этот алгоритм, чтобы **свойства** width и height задавали размеры не содержимого, а размеры блока.

10. Какие значения можно задать для свойства *box-sizing*?

***content-box***

Задаёт ширину и высоту только для содержимого  
Все остальные границы и отступы будут приплюсовываться к этой величине, делая элемент ещё шире

2. !!***padding-box***!! сообщает браузеру, что при установке для стиля свойства ширины или высоты должны включать отступы как часть своего значения

3. ***border-box*** Свойства [width](https://developer.mozilla.org/ru/docs/Web/CSS/width) и [height](https://developer.mozilla.org/ru/docs/Web/CSS/height) включают контент, внутренний отступ и границы, но не включают внешний отступ.

11. Для чего используется свойство *overflow*?

**Свойство** CSS **overflow** определяет, необходимо ли для переполненного блочного элемента содержимое обрезать, предоставить полосы прокрутки или просто отобразить. **Использование свойства overflow** со значениями, отличными от visible , (значение, принятое по умолчанию), создаёт новый блочный контекст форматирования.

12. Что обозначают значения в следующем объявлении ***border-color: yellow red green blue;***?

Top, right, bottom, left

13. Какое свойство можно использовать, чтобы задать ширину правой границы?

border-rigth:10px;