

Практическая работа

Тема: «CSS. Контейнеры. Позиционирование»

Цель: Ознакомиться с блочной разметкой страницы. Научиться управлять размещением блочных элементов на странице.

Теоретическая часть

Блочная модель

Для браузера каждый тег – это контейнер, у которого есть содержимое, внутренний отступ, внешние поля, а также рамка. Блок занимает пространство на странице равное сумме ширины содержимого, отступа, поля и рамки.

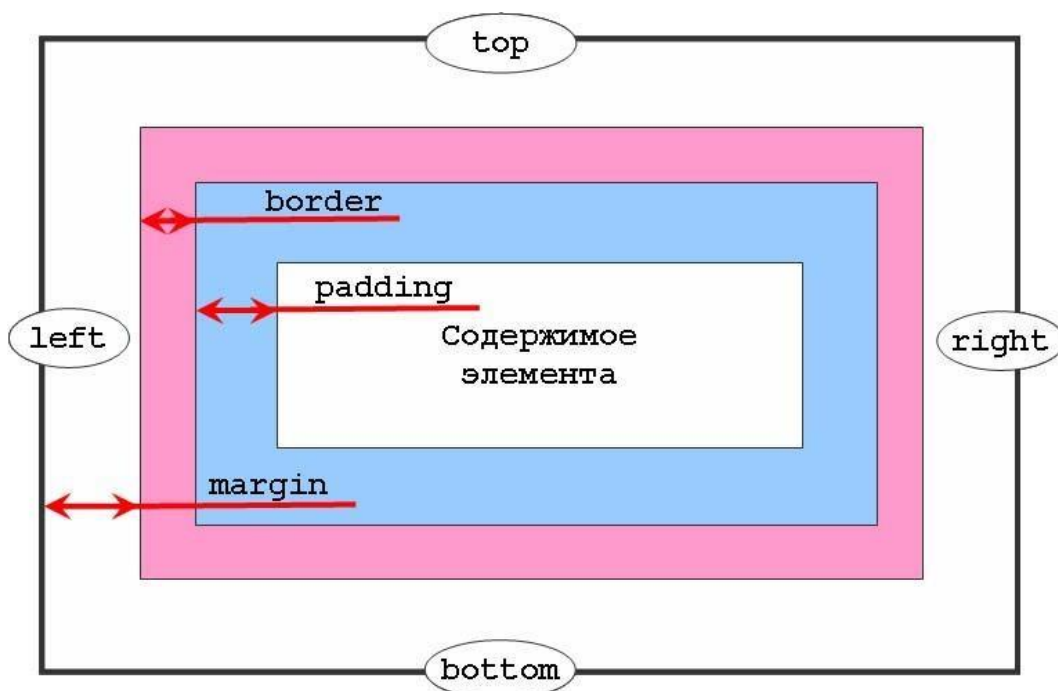


Рис. 1. Блочная модель элемента

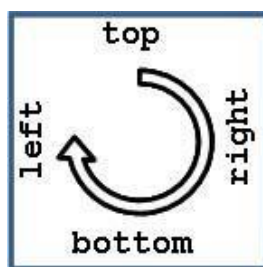


Рис. 2. Порядок установки параметров

Для всех приведенных ниже атрибутов разрешается использовать 1, 2, 3 или 4 значения, перечисляя их через пробел. В качестве значений указываются числа в любом допустимом для CSS формате. В случае применения процентов, отсчет ведется относительно ширины блока.

Атрибуты блочных элементов

Margin

Возможные атрибуты:

- margin-top - ширина верхнего поля.
- margin-right - ширина правого поля.
- margin-bottom - ширина нижнего поля.
- margin-left - ширина левого поля.
- inherit - применяется значение родительского элемента.

```
p {
margin: 20px 30px 5px 0;
}
```

padding

Возможные атрибуты:

- padding-top - ширина верхнего промежутка.
- padding-right - ширина правого промежутка.
- padding-bottom - ширина нижнего промежутка.
- padding-left - ширина левого промежутка.
- inherit - применяется значение родительского элемента.

```
h1 {
padding: 10px 100px;
}
```

border

Возможные атрибуты:

- border-width - толщина границ.
- border-style - стиль границ.
- border-color - цвет границ.
- inherit - применяется значение родительского элемента.

```
p {
border: 5px groove lightgrey;
}
```



Рис. 3. Тип границ элемента

Float (обтекание блока, плавающий блок)

Возможные атрибуты:

- right - выравнивает блок по правому краю, а остальные элементы будут обтекать его с левой стороны, начиная с самого верха;
- left - выравнивает блок по левому краю, а остальные элементы будут обтекать его с правой стороны, начиная с самого верха;
- none - это значение стоит по умолчанию, блок остаётся на месте;
- inherit - наследует значение у родительского блока.

Clear (запрет на обтекание)

Возможные атрибуты:

- right - отменяет обтекание структурного блока с правой стороны;
- left - отменяет обтекание структурного блока с левой стороны;
- both - отменяет обтекание блока с двух сторон;
- none - отменяет действие данного свойства;
- inherit - наследует значение у родителя.

Ширина блочных элементов

- Если ширина не указана явно тогда она получает значение auto (width: auto) и элемент занимает всю доступную ширину контентной области родителя. В данной ситуации width родителя = реальной ширине дочернего блока.
- Изменение ширины контента родителя влечет за собой изменение ширины дочерних элементов.
- Ширина элемента не может стать меньше указанного min-width (если конечно оно задано) даже если ширина родителя окажется меньше.
- Ширина элемента не может стать больше указанного max-width даже если размеры родителя ему это позволяют.
- Внешние отступы (margin-left, margin-right) влияют на реальную ширину блока.
- Если ширина явно указана в абсолютных единицах, тогда элемент будет иметь заданную ширину независимо от ширины родителя. Внешние отступы в таком случае тоже уже не влияют на размер.
- Если ширина явно указана в относительных единицах, то она будет рассчитываться относительно ширины контентной области родителя. Будет каждый раз пересчитана при изменениях width родителя. Внешние отступы на размер элемента влияния оказывать не будут.

Высота блочного элемента

- По умолчанию высота принимает значение auto (height: auto) и зависит от реальной высоты содержимого. Соответственно, если меняется высота содержимого, меняется высота и блока.
- Если блоку заданы минимальная или максимальная высота (min-height, max-height), или высота (height) явно указана — поведение блочного элемента аналогично поведению с min-width/max-width/width.

Практическая часть

Задание 1

Поместить текст в блок. Группа Блок позволяет заключить текст в блок. Вверху обычный текст, а внизу текст после применения стиля Блок.

Обычный текст, идущий себе потихоньку строка за строкой. Обычный текст, идущий себе потихоньку строка за строкой. Обычный текст, идущий себе потихоньку строка за строкой. Обычный текст, идущий себе потихоньку строка за строкой. Обычный текст, идущий себе потихоньку строка за строкой. Обычный текст, идущий себе потихоньку строка за строкой.

Обычный текст, идущий себе потихоньку строка за строкой. Обычный текст, идущий себе потихоньку строка за строкой. Обычный текст, идущий себе потихоньку строка за строкой. Обычный текст, идущий себе потихоньку строка за строкой. Обычный текст, идущий себе потихоньку строка за строкой. Обычный текст, идущий себе потихоньку строка за строкой. Обычный текст, идущий себе потихоньку строка за строкой.

Содержимое файла HTML

```
<HTML>
```

```
<HEAD>
```

```
<TITLE> DIV </TITLE>
```

```
<LINK rel="stylesheet" type="text/css" href="styles.css">
```

```
</HEAD>
```

```
<BODY>
```

Обычный текст, идущий строка за строкой. Обычный текст, идущий строка за строкой. Обычный текст, идущий строка за строкой. Обычный текст, идущий строка за строкой. Обычный текст, идущий строка за строкой. Обычный текст, идущий строка за строкой.

```
<div>
```

Обычный текст, идущий строка за строкой. Обычный текст, идущий строка за строкой. Обычный текст, идущий строка за строкой. Обычный текст, идущий строка за строкой. Обычный текст, идущий строка за строкой. Обычный текст, идущий строка за строкой.

```
</div>
```

```
</BODY>
```

```
</HTML>
```

Содержимое файла CSS

```
div {  
    width: 200px; /* Ширина */  
    background: #fc0; /* Цвет фона */  
    padding: 20px; /* Поля */  
}
```

Задание 2.

Создать плавающие элементы по образцу. "Плавающие" и не "плавающие" объекты. Задайте фоны и границы.



Содержимое файла HTML

```
<HTML>
```

```
<HEAD>
```

```
<TITLE> DIV </TITLE>
```

```
<LINK rel="stylesheet" type="text/css" href="styles.css">
```

```
</HEAD>
```

```
<BODY>
```

```
<div> Неплавающий элемент </div>
```

```
<br>
```

```
<div> Неплавающий элемент </div>
```

```
<br>
```

```
<div> Неплавающий элемент </div>
```

```

<br>
<div class="layer1"> Плавающий элемент </div>
<div class="layer1"> Плавающий элемент </div>
<div class="layer1"> Плавающий элемент </div>

```

```

</BODY>
</HTML>

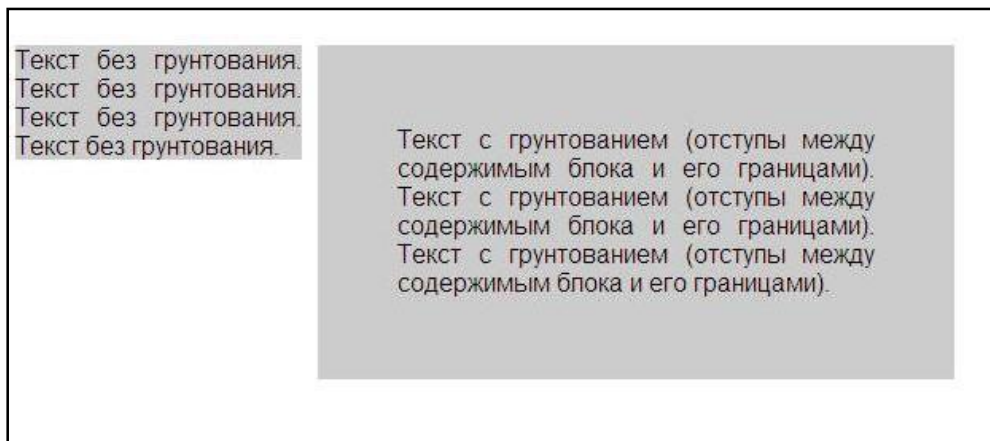
```

Содержимое файла CSS

```
.layer1 {float: left; background: #40E0D0; width: 392px}
```

Задание 3.

Задание внутренних отступов в блоке. Параметры отступа - грунтования (Padding). Слева обычный блок, справа - со значениями Padding (грунтование 50 пикселей со всех сторон). Задайте фон.



Содержимое файла HTML

```

<HTML>
<HEAD>
<TITLE> DIV </TITLE>
<LINK rel="stylesheet" type="text/css" href="styles.css">
</HEAD>
<BODY>
<div id=b11> Текст без грунтования. Текст без грунтования. Текст без грунтования. </div>
<div id=b12 > Текст с грунтованием. Текст с грунтованием. Текст с грунтованием. Текст с грунтованием. Текст
с грунтованием. Текст с грунтованием. Текст с грунтованием. </div>

</BODY>
</HTML>

```

Содержимое файла CSS

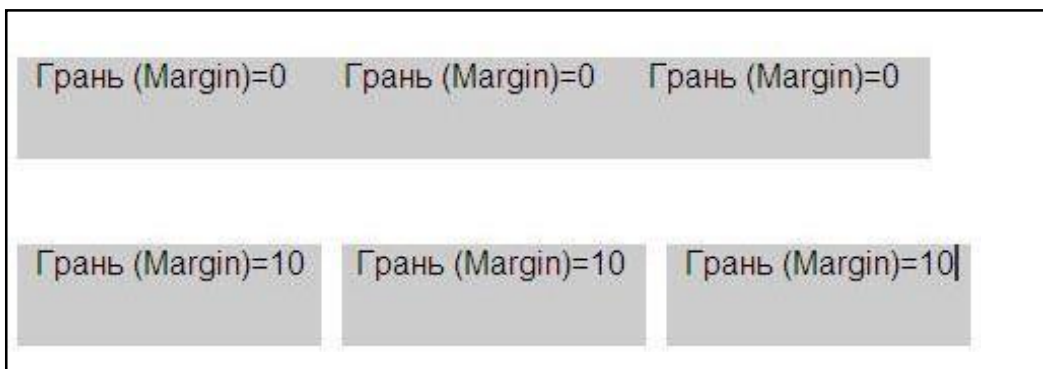
```

#b11 {float: left}
#b12 {float: left; padding-top: 50px; padding-right: 70px; padding-bottom: 60px; padding-left: 30px }

```

Задание 4.

Работа с внешними отступами (margin). Параметры **Отступ** (Margin) определяют отступы между внешними границами блока и его содержимым. Вверху - без отступа, внизу со значениями отступа 10 пикселей справа. Добавьте границы в каждом блоке и измените фоны.



Содержимое файла HTML

```
<HTML>
<HEAD>
<TITLE> DIV </TITLE>
<LINK rel="stylesheet" type="text/css" href="styles.css">
</HEAD>
<BODY>
<p class="n"> Грань(Margin)=0
<p class="n"> Грань(Margin)=0
<p class="n"> Грань(Margin)=0
<br>
<p class="y"> Грань(Margin)=10
<p class="y"> Грань(Margin)=10
<p class="y"> Грань(Margin)=10

</BODY>
</HTML>
```

Содержимое файла CSS

```
.n {float: left; background: #40E0F0}
.y { background: #40E0F0; float: left; margin-top: 10px; margin-right: 10px; margin-bottom: 10px; margin-left: 10px}
```

Также разметку сайта можно создать с помощью блочного каркаса сайта (каждый элемент – отдельный блок div)

