

# Верстка web-страниц

Поля, отступы, границы

Титова Ольга Ивановна  
Минск, 2017



# Содержание

1. Понятие блочной модели
2. Padding
3. Margin
4. Конфликты полей
5. Отрицательные значения
6. Добавление границ
7. Скругленные углы
8. Тени



# Блочная модель

**Блочная модель** представляет собой основу выполнения верстки и дальнейшего применения стилевых таблиц

Браузер обрабатывает теги как различные блоки, для него любой тег – это контейнер с содержимым.

Из формирования расположения этих блоков на странице и задания им определенных стилевых свойств и состоит процесс создания веб-страницы



# Блочная модель

**Padding** – отступ, заполнение – промежуток между содержимым и его границей

**Border** – граница, рамка – линия, черта, контур с любой стороны элемента

**Background-color** – цвет фона –заполняет пространство внутри границы, включая область padding

**Margin** – поле – это пустое пространство, которое отделяет один тег от другого



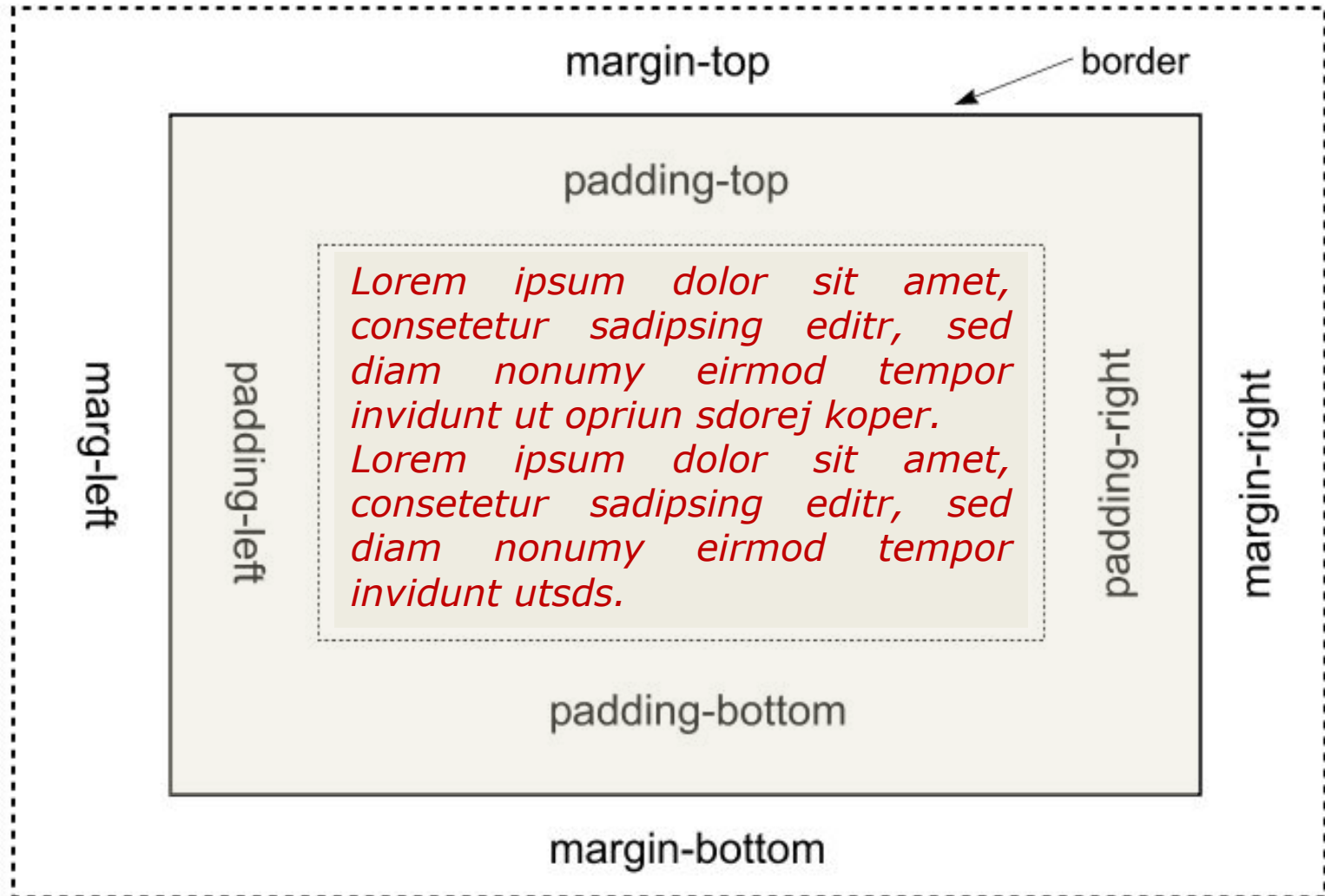
# Блочная модель

## **ВАЖНО:**

Учитывая тот факт, что разные браузеры по-разному экспериментируют с базовыми настройками отступов, то лучше их «обнулять» перед написанием своих стилей



# Пример



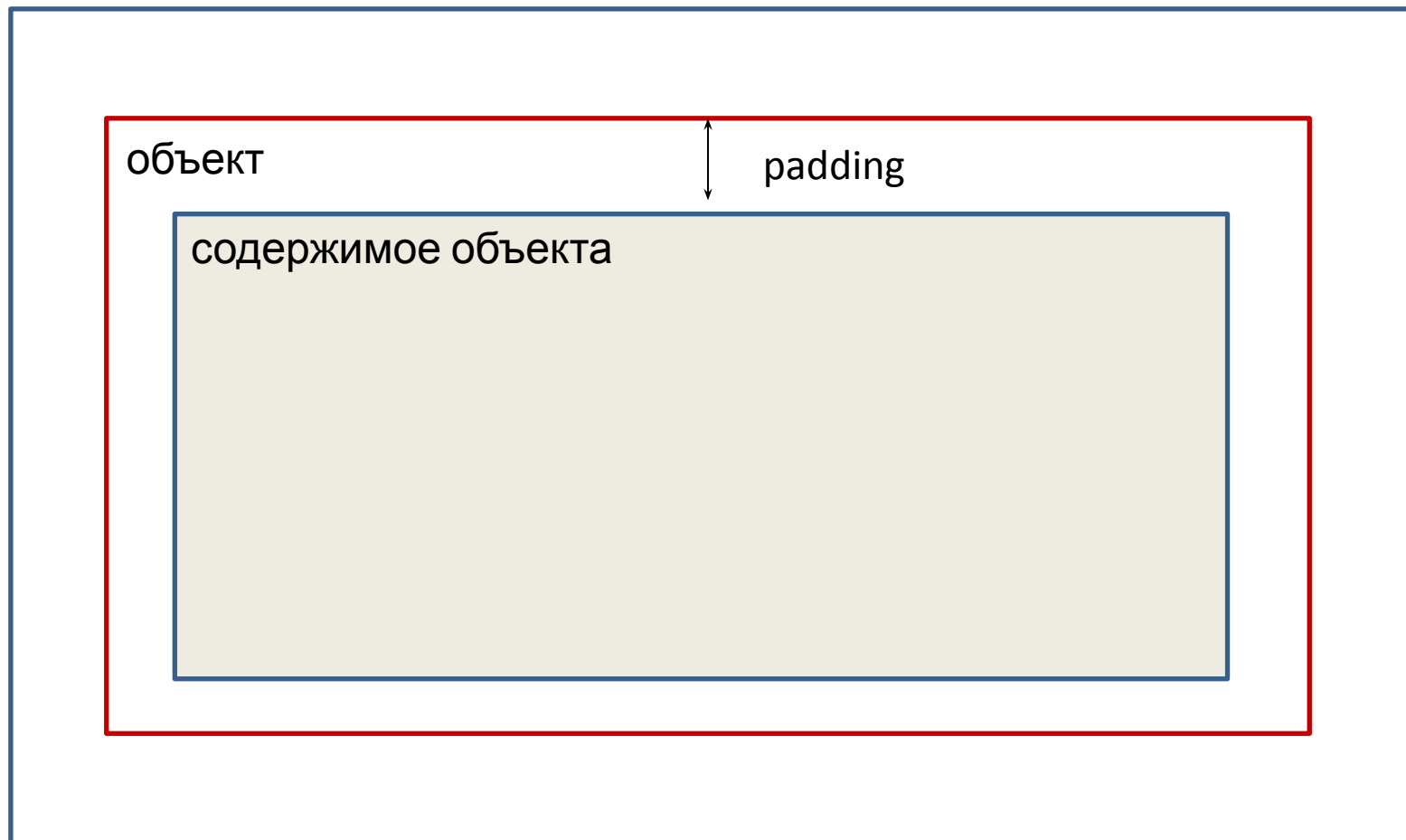
# Padding

**Padding** - устанавливает значение полей от края элемента до его содержимого. Полем называется расстояние от внутреннего края рамки элемента до воображаемого прямоугольника, ограничивающего его содержимое.

Свойство `padding` позволяет задать величину поля сразу для всех сторон элемента или определить ее только для указанных сторон.



# Padding





# Пример

*внутренние отступы для элемента;  
в данном примере в роли элемента  
выступает абзац (<p>);  
отступ между условной границей абзаца  
(условный прямоугольник) и его  
содержимым (самим текстом)*

padding



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam lectus nibh, auctor ac, vulputate in, vulputate ac, est. Donec hendrerit aliquet ipsum. Donec id nibh. Nullam ornare, magna id accumsan suscipit, lorem nisl ullamcorper tortor, in viverra leo neque ut lacus.



# Пример

```
p  
{  
padding:5px;  
}
```

со всех сторон абзаца отступ до текста будет в 5 пикселей

```
p  
{  
padding: 10px 5px 15px 3px;  
}
```

если задать четыре значения через пробел, то первый отступ – сверху,  
второй – справа и дальше по часовой стрелке



# Padding

`padding-left`

устанавливает отдельно отступ слева

`padding-right`

устанавливает отдельно отступ справа

`padding-top`

устанавливает отдельно отступ сверху

`padding-bottom`

устанавливает отдельно отступ снизу



# Пример

```
p  
{  
padding-left: 5px;  
}
```

с левой стороны абзаца отступ до текста будет в 5 пикселей

```
p  
{  
padding-top: 7px;  
}
```

сверху от абзаца отступ до текста будет в 7 пикселей



# Margin

**Margin** - устанавливает величину отступа от края элемента до соседних с ним элементов. Отступом является пространство от границы текущего элемента до внутренней границы его родительского элемента.

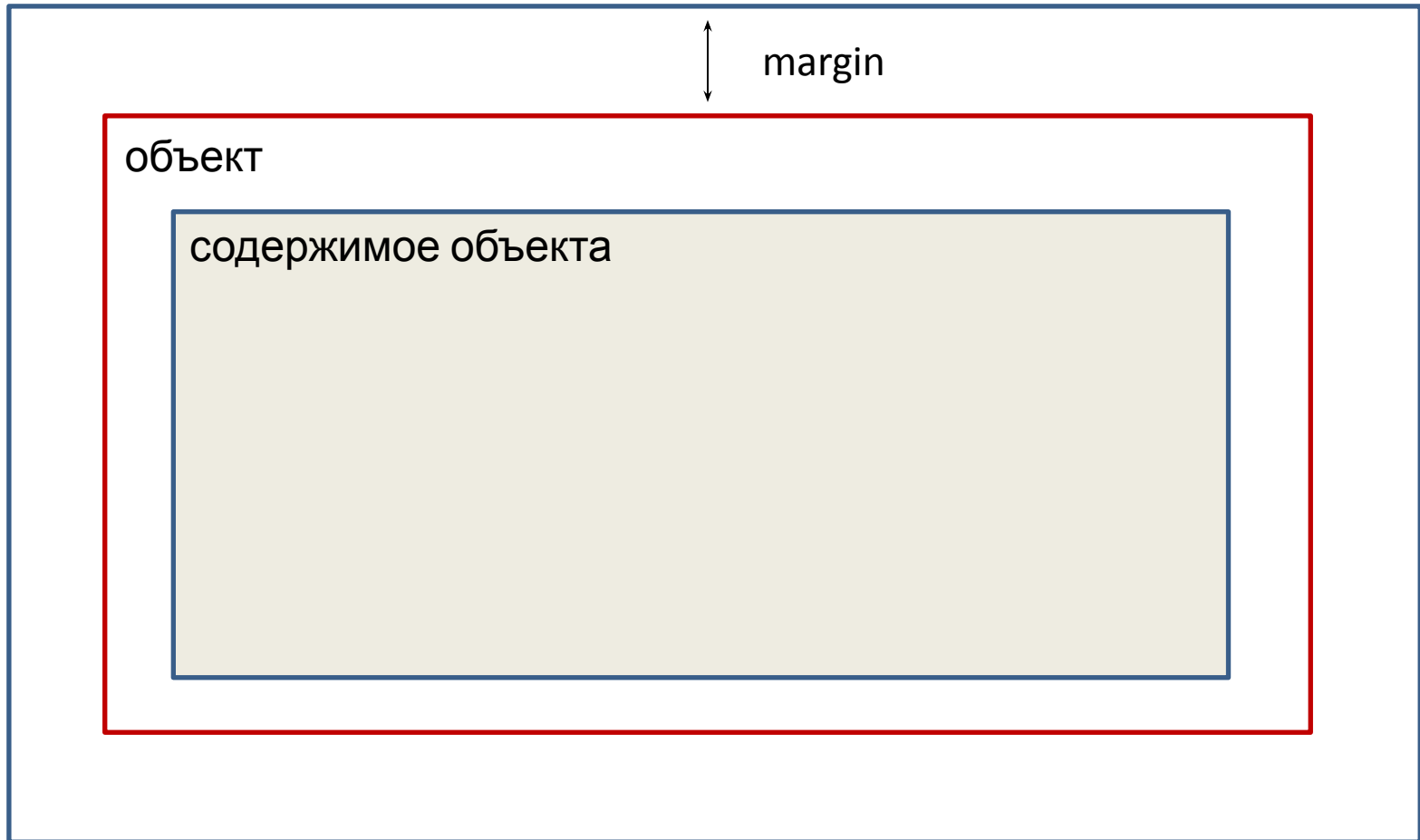
Если у элемента нет родителя, отступом будет расстояние от края элемента до края окна браузера с учетом того, что у самого окна по умолчанию тоже установлены отступы.

Чтобы от них избавиться, следует устанавливать значение `margin` для селектора `<body>` равное нулю.

Свойство `margin` позволяет задать величину отступа сразу для всех сторон элемента или определить ее только для указанных сторон.



# Margin



# Пример

*внешние отступы для элемента;  
в данном примере в роли элемента  
выступает абзац (<p>);  
отступ между условной границей абзаца  
(условный прямоугольник) и другими  
внешними объектами*

margin



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam lectus nibh, auctor ac, vulputate in, vulputate ac, est. Donec hendrerit aliquet ipsum. Donec id nibh. Nullam ornare, magna id accumsan suscipit, lorem nisl ullamcorper tortor, in viverra leo neque ut lacus.



# Пример

```
p  
{  
margin:5px;  
}
```

со всех сторон абзаца отступ будет в 5 пикселей

```
p  
{  
margin: 10px 5px 15px 3px;  
}
```

если задать четыре значения через пробел, то первый отступ – сверху,  
второй – справа и дальше по часовой стрелке





# Margin

`margin-left`

устанавливает отдельно отступ слева

`margin-right`

устанавливает отдельно отступ справа

`margin-top`

устанавливает отдельно отступ сверху

`margin-bottom`

устанавливает отдельно отступ снизу



# Пример

```
p  
{  
margin-left: 5px;  
}
```

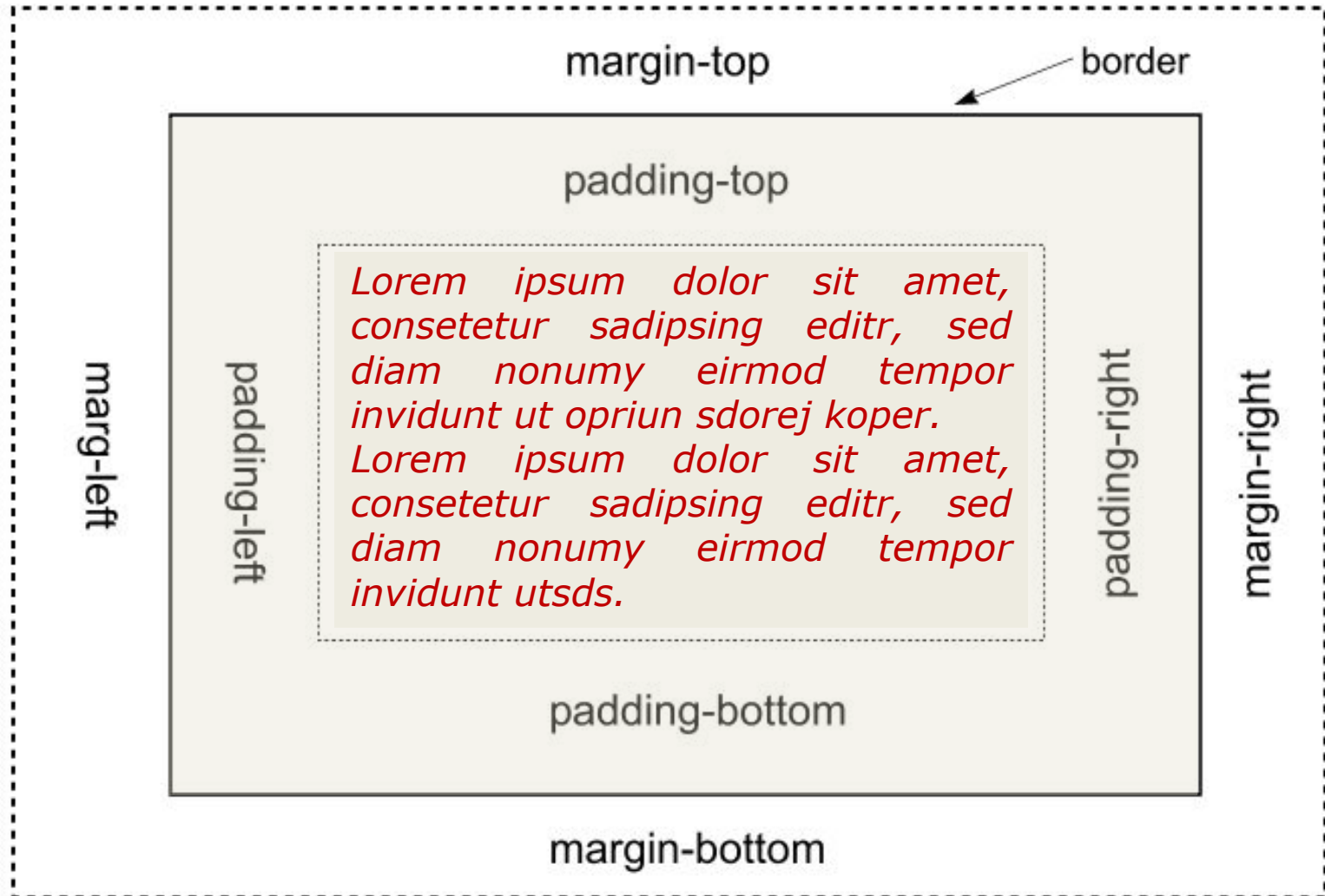
с левой стороны абзаца отступ будет в 5 пикселей

```
p  
{  
margin-top: 7px;  
}
```

сверху от абзаца отступ будет в 7 пикселей



# Пример



# Пример



Тектовое сообщение

**Введем код:**

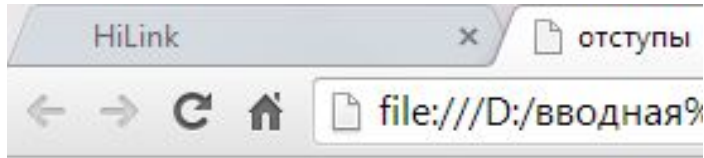
```
<!doctype html>
<html>
<head>
  <title> отступы </title>
  <link rel="stylesheet"
href="style07.css"/>
</head>
<body>


<p>Тектовое сообщение</p>

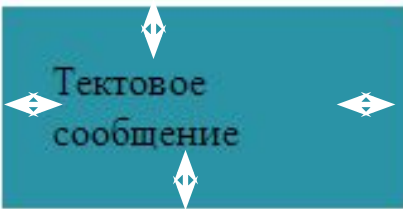
</body>
</html>
```

# Пример

Добавим описание стилей:



margin-top



padding

```
p
{
background-color:#2b93a6;
padding:20px;
width:120px;
}
```

```
img
{
margin-top:50px;
}
```

# Отступы

## **Вариант сокращенной формы:**

Если нужно применить одинаковое значение свойства поля или отступа сверху и снизу и одинаковое значение для левого и правого полей, то можно указывать только два значения

**margin: 0 2em;**

Удаляет верхнее и нижнее поля, а для правого и левого задает одинаковые значения



# Конфликты полей

## **ВАЖНО:**

В CSS не всегда справедливы стандартные математические расчеты, что обязательно необходимо учитывать.

При наличии верхнего и нижнего полей соприкасаемых элементов, общий отступ (внешний) между ними **не суммируется**, а **берется больший** из них.

**При вложенных элементах** – добавленные поля для вложенного элемента отобразятся не снизу\сверху, а суммарно сверху.



# Конфликты полей

**Горизонтальные поля не конфликтуют**

**Между относительно и абсолютно позиционируемыми элементами также нет конфликтов**





# Отрицательные значения

Отступы должны иметь **положительные** значения, но **CSS допускает использование отрицательных значений** для достижения определенных визуальных эффектов.



# Добавление границ

**Граница** представляет собой линию, которая обрамляет стилизуемый элемент (см. примеры из лекции по изображениям)

Горизонтальную линию (не привязанную к обрамлению объекта) можно создать с помощью тега `<hr>`



# Добавление границ

**width** – толщина линии может быть задана не только какой-либо единицей измерения, но и с помощью ключевых слов:

**thin** – тонкая линия

**medium** – средней толщины

**thick** – толстая линия



# Добавление границ

**style** – стиль, который управляет типом границы

**Ключевые слова** для определения данного значения:

**solid** – сплошная линия

**dashed** – штриховая (пунктирная) линия

**none\hidden** – границы удаляются

**Dotted**

**Double**

**Groove**

**Ridge**

**Inset**

**Outset**



# Добавление границ

Допустимы всевозможные **комбинации** при задании стилевых свойств:

**border-right-width**

**border-left-style**

...

**ВАЖНО:**

При написании стилевых параметров, стремимся уменьшать объем кода



# Пример

## Пример 1

border: 2px solid red;

border-bottom: 4px dashed #333;

## Пример 2

border: 2px solid red;

border-left: none;

## Пример 3

border: 2px solid green;

border-right-color: blue;

## Пример 4

border: 2px solid green;

border-color: green yellow red blue;



# Задание

Проверьте на произвольных элементах web-страницы реализацию различных значений и их комбинаций для границ



# Скругленные углы

## **ВАЖНО:**

Веб-браузеры рассматривают все элементы как прямоугольные блоки – это становится видимым при задании таких стилевых свойств как граница или фоновый цвет

Стилевое **свойство border-radius** позволяет выполнить эффект скругления углов

Чтобы увидеть данный эффект элемент должен иметь либо границу либо фоновый цвет\изображение





# Скругленные углы

## Пример

```
.specialBox {border-radius: 20px;}
```

При использовании единственного значения браузер рисует закругления одинакового радиуса для каждого угла элемента

```
.specialBox {border-radius: 0 30px 10px 5px;}
```

При использовании последовательности значений браузер рисует закругления указанных радиусов для каждого угла элемента:  
левый верхний и дальше по часовой стрелке



# Скругленные углы

## Пример

```
.specialBox {border-radius: 20px 10px;}
```

При указании двух значений – первое применяется к левому верхнему и правому нижнему углам (по диагонали), а второе – к правому верхнему и левому нижнему (по диагонали) углам



# Скругленные углы

Можно создавать не только скругленные углы, но и углы **эллиптической формы**

Для этого необходимо задать **двойное значение радиуса**

**border-radius: 40px/20px;**

1-е значение для левого(правого) края (радиус)

2-е значение для верхнего (нижнего) края (радиус)



# Скругленные углы

Доступны для использования всевозможные комбинации:

```
border-top-left-radius: 10px;
```

```
border-bottom-left-radius: 5px;
```



# Скругленные углы

Для некоторых браузеров данное свойство требует **наличие префикса производителя.**

## **-webkit-**

- применяется Chrome, Safari и др. браузерами на базе WebKit

## **-moz-**

- используется Mozilla Firefox

## **-o-**

- применяется Opera

## **-ms-**

- используется MS Internet Explorer



# Добавление теней

**Для добавления теней** к блоку можно использовать **свойство box-shadow**

1-е значение задает горизонтальное смещение (положительное значение – вправо, отрицательное - влево);

2-е значение задает вертикальное смещение (положительное – вниз, отрицательное - вверх);

3-е значение задает радиус размытия тени;

4-е значение определяет цвет тени.



# Добавление теней

Для размытости тени нельзя использовать **%**.

Значение **0** будет определять тень с четкими краями, чем выше значение, тем более размытой и тусклой становится тень.

Можно задать **одновременно несколько** теней (через запятую).

Дополнительное значение – **inset** – создаст внутреннюю тень

```
box-shadow: inset 4px 4px 8px #56f231;
```

Значение расширения – **spread** – приведет к расширению тени на указанное значение

```
box-shadow: inset 4px 4px 8px 10px #56f231;
```

Тень по горизонтали и вертикали станет больше и размытие начнется только после расширения тени



Спасибо за внимание

