# Лекция: Блочная модель (Box Model)

#### МДК 09.01. Проектирование и разработка веб-приложений

**Цель лекции:** Сформировать у студентов полное и практическое понимание блочной модели CSS, ее компонентов и их влияния на расчет итоговых размеров элемента. Научить управлять размерами и расположением элементов на странице.

## План лекции:

- 1. Что такое Box Model и зачем она нужна?
- 2. Компоненты Box Model: content, padding, border, margin.
- 3. Расчет итоговых размеров элемента.
- 4. Нюансы и "подводные камни": свойство box-sizing.
- 5. Практические примеры и разбор типичных ситуаций.
- 6. Резюме.

# 1. Что такое Box Model и зачем она нужна?

Каждый элемент в HTML-документе, будь то заголовок <h1>, параграф , изображение <img> или контейнер <div>, браузер представляет в виде прямоугольного контейнера — "бокса" (коробки).

**Блочная модель (Box Model)** — это концепция в CSS, которая описывает, как устроен этот прямоугольный контейнер. Она определяет, из каких частей он состоит, как эти части взаимодействуют друг с другом и как влияют на общие размеры элемента и его расположение относительно других элементов.

#### Почему это важно?

- **Контроль над версткой:** Без понимания Box Model верстка превращается в угадывание, почему элементы имеют не те размеры, почему они не становятся рядом или, наоборот, наезжают друг на друга.
- Предсказуемость: Вы сможете точно рассчитывать, сколько места на экране займет ваш
- Адаптивность: Понимание модели необходимо для создания гибких и отзывчивых макетов.

## Аналогия: Представьте картину на стене:

- content это само полотно с изображением.
- padding это паспарту (рамка) между картиной и рамой.
- border это сама физическая рама картины.
- margin это свободное пространство (отступ) между этой рамой и соседними картинами или стеной.

#### 2. Компоненты Box Model

Блочная модель состоит из четырех концентрических областей, расположенных изнутри наружу.

#### 2.1. Content (Содержимое)

Это ящик, внутренняя область элемента, где отображается его основной контент: текст, изображения, видео и т.д.

- Управление размерами: Ширина и высота контента задаются свойствами width и height.
- По умолчанию: Значения width и height задают размеры именно области контента.

#### Пример:

```
.box {
  width: 200px;
  height: 100px;
  background-color: lightblue;
}
```

В этом случае область с голубым фоном будет иметь размер 200х100 пикселей.

## 2.2. Padding (Внутренний отступ)

Это пространство между контентом и границей элемента. Padding "раздвигает" контент от краев, создавая внутреннее "поле". Фон элемента (заданный через background-color или background-image) распространяется на область padding.

- Управление: Свойства padding-top, padding-right, padding-bottom, padding-left.
- **Сокращенная запись:** padding: 10px 20px 15px 5px; (по часовой стрелке: верх, право, низ, лево).

```
    padding: 20px; — со всех сторон 20px.
    padding: 10px 20px; — верх/низ: 10px, право/лево: 20px.
```

#### Пример (продолжим предыдущий):

```
.box {
  width: 200px;
  height: 100px;
  background-color: lightblue;
  padding: 20px; /* Добавили внутренний отступ */
}
```

Теперь общие визуальные размеры "голубого блока" станут больше, так как к 200х100рх добавится по 20рх с каждой стороны.

# 2.3. Border (Граница)

Это линия (или более сложный стиль), которая окружает padding и контент. Граница всегда лежит между padding и margin.

• Управление: Для границы нужно задать три параметра:

```
    border-width: Толщина (например, 2px).
    border-style: Стиль (solid, dotted, dashed, double и др.).
    border-color: Цвет.
```

- Сокращенная запись: border: 2px solid red;.
- Отдельные стороны: Можно управлять каждой стороной отдельно: border-top, border-bottom, border-left, border-right.

#### Пример (продолжим):

```
.box {
  width: 200px;
  height: 100px;
  background-color: lightblue;
  padding: 20px;
  border: 5px solid black; /* Добавили черную сплошную границу */
}
```

Граница добавится внешним контуром к уже существующим размерам.

## 2.4. Margin (Внешний отступ)

Это внешнее пространство вокруг элемента. Оно отделяет данный элемент от его соседей. **Важно:** Фон элемента HE распространяется на margin.

- Управление: Свойства margin-top, margin-right, margin-bottom, margin-left.
- Сокращенная запись: Аналогична padding (например, margin: 10px;).
- Особые значения:
  - o margin: O auto; очень частый прием для центрирования блочного элемента по горизонтали.
  - o margin-left: auto; прижмет элемент к правому краю.

#### Пример (финальный для этого раздела):

```
.box {
  width: 200px;
  height: 100px;
  background-color: lightblue;
  padding: 20px;
  border: 5px solid black;
  margin: 30px; /* Добавили внешний отступ от соседей */
}
```

Теперь вокруг нашего элемента с границей появится прозрачная "буферная зона" в 30 пикселей, отталкивающая другие элементы.

## 3. Расчет итоговых размеров элемента

Это самый важный практический вывод. По умолчанию (c box-sizing: content-box;) свойства width и height задают размер только области content.

#### Формула для расчета полной ширины и высоты элемента:

- Полная Ширина = width + padding-left + padding-right + border-left-width + border-right-width + margin-left + margin-right
- Полная Высота = height + padding-top + padding-bottom + border-top-width + borderbottom-width + margin-top + margin-bottom

#### Применим к нашему примеру:

```
    width: 200px
    padding-left + padding-right: 20px + 20px = 40px
    border-left + border-right: 5px + 5px = 10px
    margin-left + margin-right: 30px + 30px = 60px
```

Итоговая ширина, которую элемент займет на странице: 200px + 40px + 10px + 60px = 310px.

Обратите внимание: визуально (включая фон и границу) элемент займет 200px + 40px + 10px = 250px. А margin — это прозрачное пространство вокруг.

# 4. Нюансы и "подводные камни": свойство box-sizing

Исторически сложилось, что расчет размеров по умолчанию (content-box) не всегда интуитивно понятен. Часто дизайнер в макете указывает ширину блока 300рх, имея в виду всю его "шапку" целиком, включая padding и border. Но верстальщику приходится вычитать: width = 300px - padding - border.

Чтобы решить эту проблему, было введено свойство box-sizing.

- box-sizing: content-box; (значение по умолчанию).
  - width и height задают размер только контента. padding и border добавляются сверху.
- box-sizing: border-box; (рекомендуемое к использованию).
  - width и height задают общий размер элемента **включая** padding и border! Область контента внутри будет автоматически подстраиваться.

## Сравним на примере:

```
.box-content {
  box-sizing: content-box;
  width: 200px;
  padding: 20px;
  border: 5px solid black;
  /* Финальная ширина: 200 + 40 + 10 = 250px */
```

```
.box-border {
  box-sizing: border-box;
  width: 200px;
  padding: 20px;
  border: 5px solid black;
  /* Финальная ширина: 200px! */
  /* Ширина контента внутри автоматически станет: 200 - 40 - 10 = 150px */
}
```

**Примечание:** Всегда в начале CSS-файла задавайте правило для всех элементов. Это избавит от множества проблем с расчетами.

```
* {
   box-sizing: border-box;
}
```

# 5. Практические примеры и разбор типичных ситуаций

#### Пример 1: Создание карточки товара.

```
.product-card {
  width: 250px; /* Фиксированная ширина карточки */
  box-sizing: border-box; /* Чтобы padding не ломал размеры */
  padding: 15px; /* Воздух внутри карточки */
  border: 1px solid #ddd; /* Тонкая серая граница */
  border-radius: 8px; /* Закругленные углы */
  margin: 10px; /* Отступ между карточками */
  background-color: white;
  text-align: center;
}
```

#### Пример 2: "Схлопывание" внешних отступов (Margin Collapsing).

Это важное поведение, когда два вертикальных margin соседних блоков сливаются в один, равный наибольшему из них.

```
<div class="block1">Блок 1</div>
<div class="block2">Блок 2</div>
```

```
.block1 {
  margin-bottom: 50px;
  background: red;
}
.block2 {
  margin-top: 30px;
  background: blue;
}
```

Расстояние между блоками будет не 50px + 30px = 80px, а max(50px, 30px) = 50px.

# Пример 3: Центрирование блока по горизонтали.

```
.container {
  width: 80%; /* или любая другая ширина */
  max-width: 1200px;
  margin: 0 auto; /* "волшебная" строка для центрирования */
}
```

# 6. Резюме

- 1. **Box Model** это фундаментальная концепция CSS, описывающая структуру каждого элемента на странице как набор концентрических прямоугольников: content, padding, border, margin.
- 2. Понимание модели критически важно для точного позиционирования элементов и управления макетом.
- 3. **По умолчанию** свойства width и height задают размер только области content. padding и border добавляются к этим размерам.
- 4. Полная ширина/высота элемента рассчитывается как сумма width/height, padding, border и margin.
- 5. Свойство box-sizing: border-box меняет логику расчета, заставляя width/height включать в себя padding и border. Это гораздо более удобный и предсказуемый подход, который рекомендуется использовать повсеместно.
- 6. Следует помнить о таких особенностях, как **схлопывание вертикальных отступов (margin collapsing)**.

## Контрольные вопросы:

- 1. Назовите и охарактеризуйте все компоненты Box Model.
- 2. Элемент имеет width: 100px, padding: 10px и border: 5px solid. Какова его итоговая ширина при box-sizing: content-box и при box-sizing: border-box?
- 3. Как центрировать блочный элемент по горизонтали?

4. Зачем нужно свойство box-sizing?