

## Построение сеток на флексах по макету

Выполнен на 0%

Введение

Углублённая теория

- Свойство display и управление потоком документа
- Алгоритм раскладки на флексах
- Инструменты для работы с флексбоксом

Методики работы с флексами

Кейс 1, лёгкий уровень

Кейс 2, лёгкий уровень

Кейс 3, лёгкий уровень

Кейс 4, лёгкий уровень

Кейс 5, средний уровень

Кейс 6, средний уровень

Кейс 7, средний уровень

Кейс 8, сложный уровень

Кейс 9, сложный уровень

Кейс 10, сложный уровень

Главная / Мое обучение / Построение сеток на флексах по макету / Углублённая теория /

# Свойство display и управление потоком документа

Каждому HTML-элементу на странице соответствует прямоугольная область. Такие области называются **боксами**.

По умолчанию все HTML-элементы обладают определёнными боксами. Но стандартный тип отображения бокса можно изменить с помощью CSS-свойства `display`. Тип отображения может влиять сразу на две вещи: на расположение самого бокса в потоке и на расположение дочерних элементов внутри бокса.

Например, у элемента `<li>` тип бокса по умолчанию — `display: block`. Это означает, что элементы списка отображаются один под другим на странице. Если изменить тип отображения на `inline`, элементы списка будут отображаться рядом друг с другом, как будто это слова в предложении.

Тот факт, что вы можете изменить значение отображения любого элемента, означает, что вы можете выбрать HTML-элементы согласно их семантическому значению, не заботясь о том, как они будут выглядеть. То, как они выглядят, можно изменить.

### Значение

Несмотря на то, что тип отображения можно изменить, всё-таки лучше использовать элементы учитывая их поведение в потоке. К примеру, для отдельно стоящего текстового блока использовать `<p>`, а не `<span>` с заданным значением `display: block`.

Рассмотрим четыре основных типа боксов:

- блочные, `display: block`,
- строчные, `display: inline`,
- блочно-строчные, `display: inline-block`,
- гибкие или флексовые (флексы), `display: flex`.

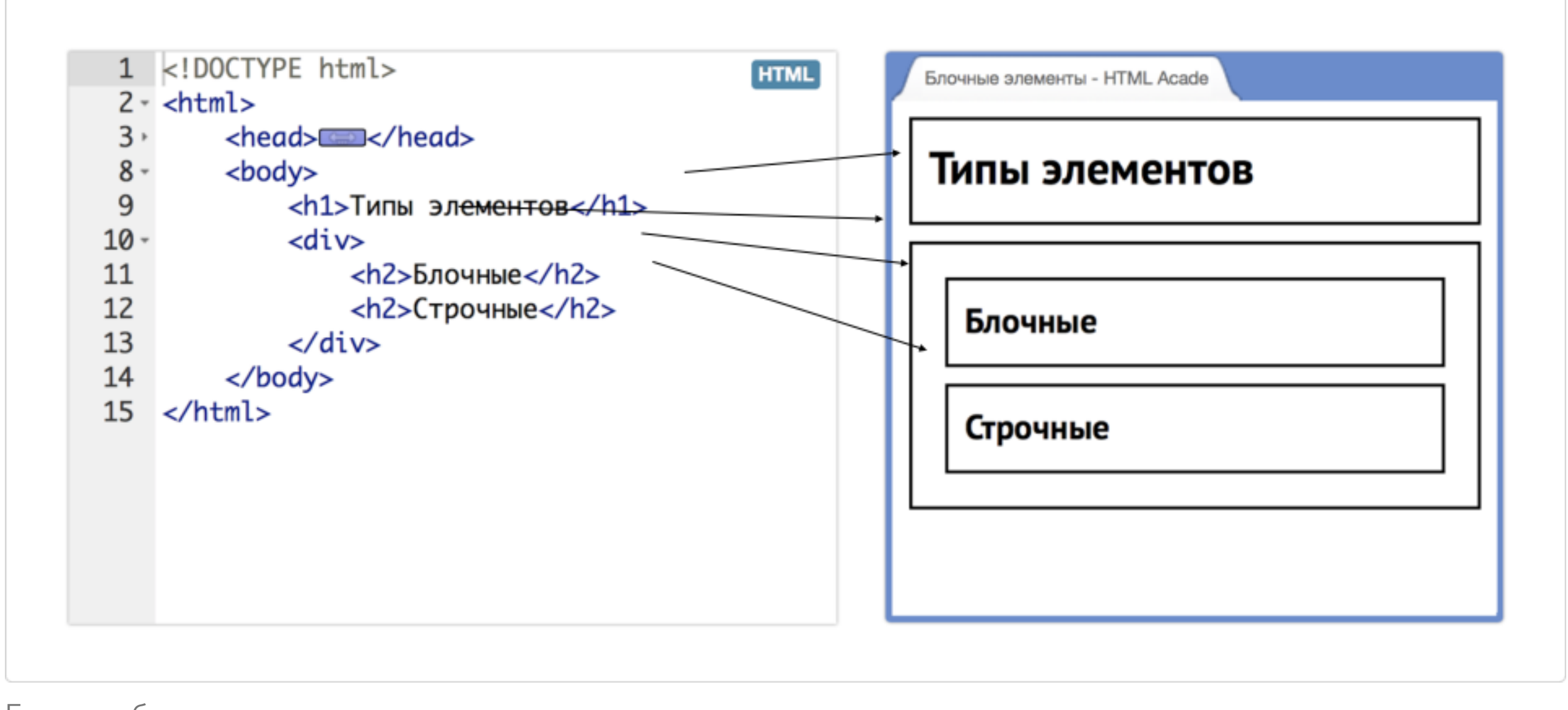
## Блочный бокс

Блочный бокс — это прямоугольная область на странице, просто прямоугольник. По умолчанию блочным боксом обладают крупные поточные элементы, которые не являются фразовыми. Например, элементами с блочными боксами по умолчанию являются: `<div>`, `<header>`, `<footer>`, `<section>`, `<h1>` ..., `<h6>`, `<p>`, `<ul>`, `<ol>`, `<li>`.

У блочных боксов есть важные особенности поведения:

- Блочный бокс «эгоистичен». Это выражается в том, что до и после него есть принудительный перенос строки. На одной строке с таким боксом не может находиться никакой другой элемент. Даже если блочный бокс почти не содержит контента или занимает лишь маленькую часть строки — он всё равно на строке один.
- На блочные боксы нормально действуют все свойства блочной модели (`width`, `height`, `margin`, `padding`).
- Блочный бокс по умолчанию тянется на всю ширину родительского элемента.
- По высоте же блочные боксы ужимаются под содержимое.

Рассмотрим на примере, как себя ведут блочные боксы.



Блочные боксы

Внутри `<body>` есть четыре тега, и все эти теги являются блочными.

Хорошо видно, что строки текста внутри `<h2>` очень короткие, и можно было бы эти элементы ужать под текст. Но заголовки тянутся на всю ширину и занимают всю доступную в родителе ширину, за исключением внутренних отступов.

Также очевидно, что если бы заголовки ужились под контент, то они поместились бы на одну строку. Но каждый занимает отдельную строку.

## Строчный бокс

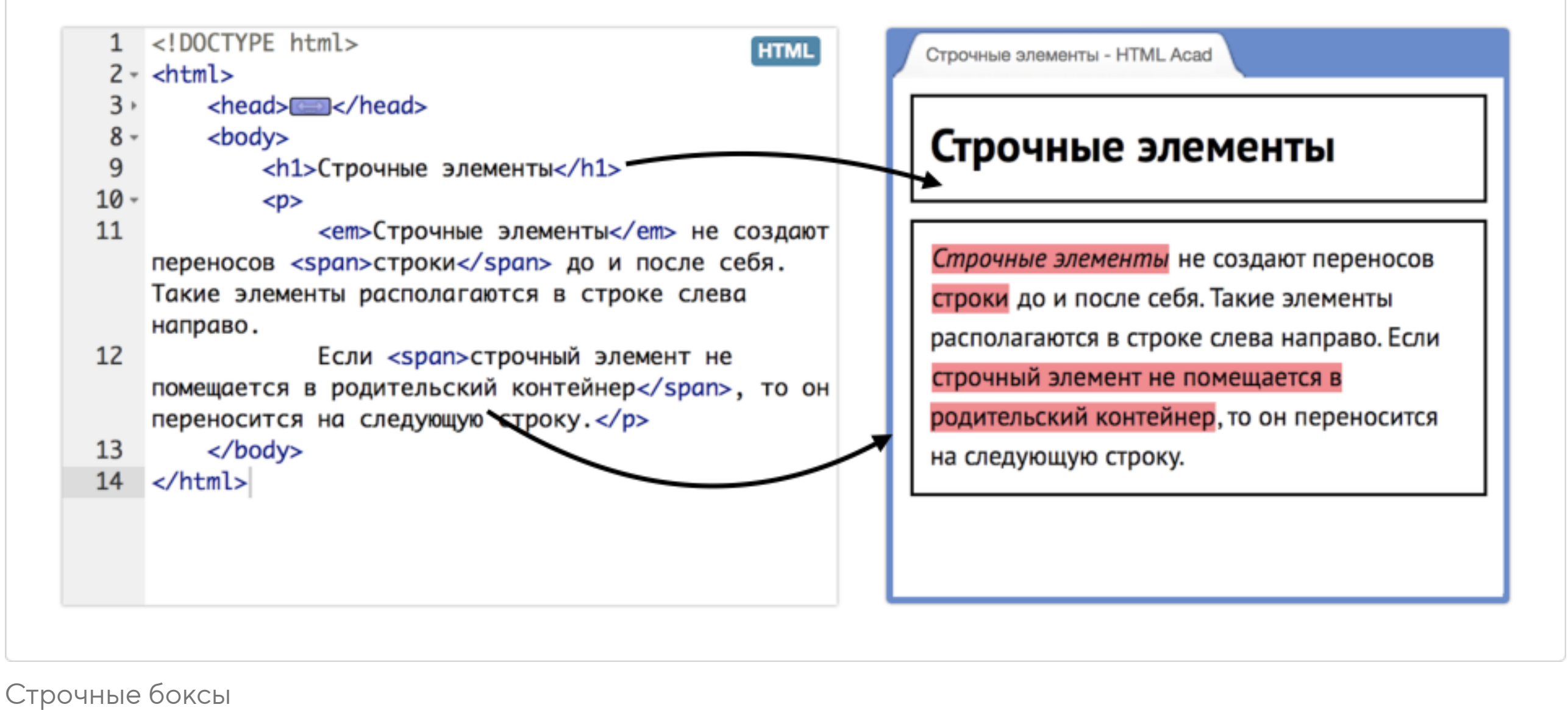
Строчные боксы — противоположность блочных. Если блочный бокс — это крупная неразрывная прямоугольная область, то строчный бокс — это фрагмент текста, который может разрываться и находиться на нескольких строках.

Элементы, боксы которых по умолчанию являются строчными, это теги, с помощью которых размечают небольшие куски текста, словосочетания: `<a>`, `<span>`, `<button>`, `<strong>`, `<em>`, `<i>`, `<b>`, `<time>`.

У строчных боксов несколько отличий от блочных:

- Строчные боксы ведут себя как текст, то есть могут располагаться и на одной строке, и на нескольких. Если такой бокс не помещается на одной строке, то он спокойно переносится на следующую.
- Строчные боксы плохо взаимодействуют со свойствами блочной модели. Некоторые свойства на них просто не действуют, а некоторые — действуют частично. Например, отступы в строчных боксах работают только в горизонтальном направлении.
- Размеры строчных боксов всегда зависят от содержания.

В примере ниже, благодаря подсветке, видно как ведёт себя строчный бокс, если он не помещается в строке.



Строчные боксы

Строчные боксы — это не только теги `<em>` и `<span>`, но и обычный текст, который не обернут во фразовый тег. Такие фрагменты текста оборачиваются в анонимный строчный бокс. Анонимных боксов в рассмотренном примере — три (это участки между явными строчными боксами, которые принадлежат тегам).

Блочный бокс — это прямоугольный статичный «кирпич», а строчный бокс — это что-то гибкое и текучее внутри этого «кирпича».

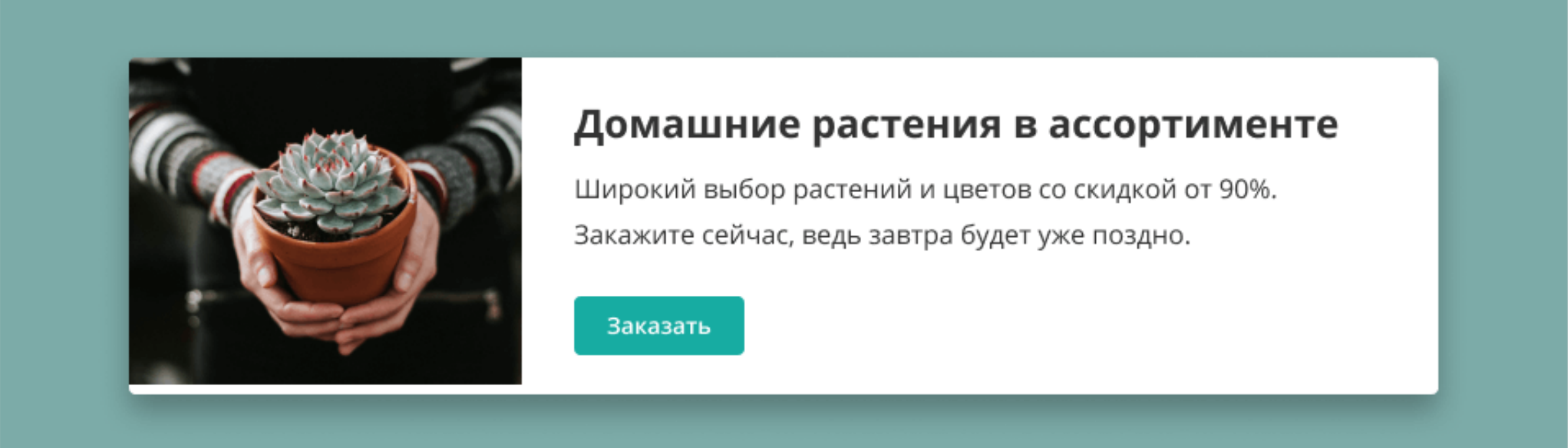
## Блочно-строчный бокс

Блочно-строчные боксы сочетают в себе особенности и блочных боксов, и строчных боксов. Этот тип боксов используются для стилизации мелких интерфейсных элементов и элементов текстового содержания.

### Особенности тега <img>

Тег `<img>` относится к строчным боксам (`display: inline`). Но в отличие от стандартных строчных боксов, ему можно задавать в CSS отступы `padding / margin`, размеры `width`, `height` (по умолчанию размеры определяются значениями встроенного изображения), параметры границы `border`, `border-radius`. То есть, его поведение схоже с блочно-строчным боксом (`display: inline-block`). В тоже время, `<img>` не имеет базовой линии (`baseline`). Если для изображения указать `vertical-align: baseline`, его нижняя часть будет помещена на базовую линию текста. Проще говоря, если изображение поместить в контейнер и не указать для него выравнивание по вертикали (по умолчанию выравнивание вдоль базовой линии), то изображение не будет плотно прилегать к нижнему краю контейнера, даже если в этом контейнере кроме изображения ничего нет. Контейнером для изображения может быть ссылка `a`, `figure`, обычный `div` и другие.

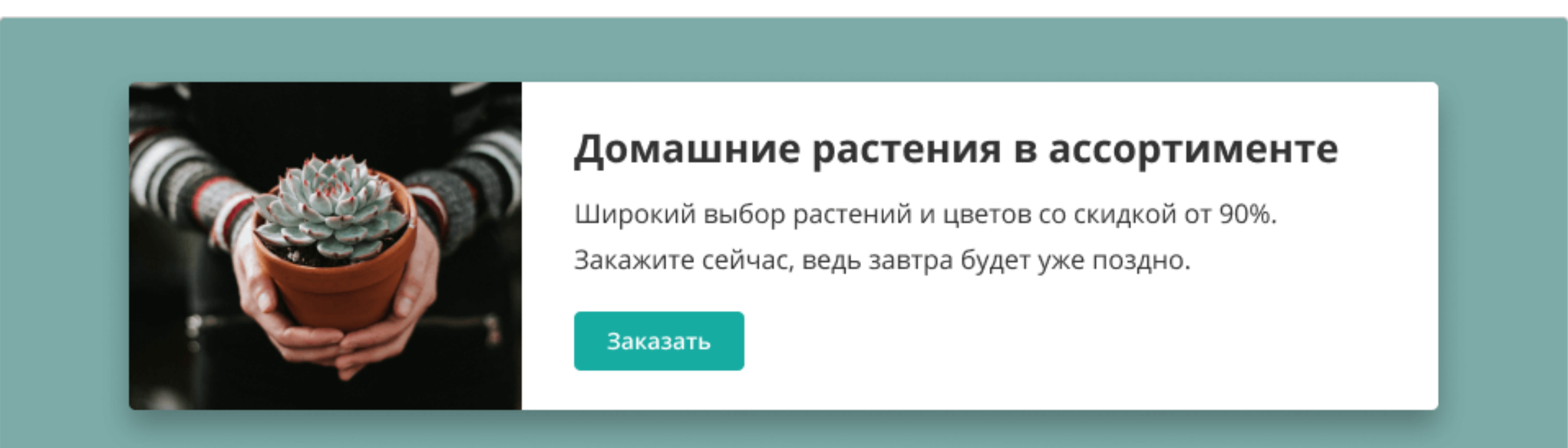
```
<section class="card">
  <figure>
    
  </figure>
  <h2>Домашние растения в ассортименте</h2>
  <p class="text">Широкий выбор растений и цветов со скидкой от 90%. Закажите сейчас, ведь завтра будет уже поздно.</p>
  <a class="button" href="#">Заказать</a>
</section>
```



Пустое пространство под изображением при использовании `vertical-align: baseline`.

Убрать пустое пространство под изображением можно с помощью `vertical-align: middle`.

```
...
.card img {
  vertical-align: middle;
}
...
```



Убираем пустое пространство под изображением с помощью `vertical-align: middle`.

Кстати, CSS-свойство `vertical-align` устанавливает вертикальное выравнивание только строчным, блочно-строчным боксам или ячейкам таблицы. Для блочных боксов это CSS-свойство не работает.

## Флексбокс

Флексы — основной инструмент построения крупных сеток и микросеток.

Есть несколько особенностей, которые следует помнить и учитывать при использовании флексов. Перечислим их:

- Все блоки очень легко делаются «резиновыми», что уже следует из названия «flex». Элементы могут сжиматься и растягиваться по заданным правилам, занимая нужное пространство.
- Выравнивание по вертикали и горизонтали, базовой линии текста отлично работает.
- Расположение элементов в HTML не имеет решающего значения. Его можно поменять в CSS.
- Элементы могут автоматически выстраиваться в несколько строк/столбцов, занимая всё предоставленное место.
- Множество языков в мире используют написание справа налево `rtl` (right-to-left), в отличие от привычного нам `ltr` (left-to-right). Флекс адаптирован для этого. В нём есть понятие начала и конца, а не права и лева. В браузерах с локально `rtl` все элементы будут автоматически расположены в реверсном порядке.
- Синтаксис CSS-правил очень прост и осваивается довольно быстро.

При этом есть вещи, которые не стоит забывать при использовании флексбокса:

- Не используйте флексы там, где в этом нет необходимости.
- Разберитесь с флексбоксом и знайте его основы. Так намного легче достичь желаемого результата. К примеру, по умолчанию флекс-элементы вытягиваются по поперечной оси и установленное значение высоты для элемента никак не учитывается.
- Не забывайте про `margin`-ы. Они учитываются при установке выравнивания по осям. Важно помнить, что `margin`-ы во флексбоксе не «схлопываются», как это происходит в обычном потоке.
- Значение `float` у флекс-блоков не учитывается и не имеет значения.

## Ознакомились со статьёй?

Сохранить прогресс

Углублённая теория

Алгоритм раскладки на флексах