

Веб- программирование: HTML + CSS

Содержание

01 Важность

02 HTML

03 CSS

04 DevTools

05 Tailwind CSS

06 SEO

07 Адаптив

08 Вопросик

HTML и CSS - база веба

Историческая справка: HTML

- HTML был создан в 1991 году Тимом Бернерсом-Ли в ЦЕРН
- Цель — упростить обмен документами через интернет, используя гипертекст для связывания информации
- Версии:
 - 1 (1991): заголовки, абзацы, списки и ссылки
 - 2 (1995) : таблицы, формы и улучшение форматирования текста
 - 3 (1997): скрипты и апплеты
 - 4 (1999): CSS, разделение контента и стилизации
 - 5 (**2014**): семантика, мультимедиа, доступность

Историческая справка: CSS

- CSS был создан в 1994 году Хоконом Вимом Ли
- Цель — сделать веб красивее
- Версии:
 - 1 (1996): стилизация фона, текста и границ
 - 2 (1998) : позиционирование, медиа-запросы и поддержка таблиц
 - 3 (2001): градиенты, тени, трансформации

HTML

- HTML – HyperText Markup Language – стандартизованный язык гипертекстовой разметки документов, который используется для создания веб-страниц в интернете.
- Фундамент веба
- Язык разметки, а не программирования
- Структурирует контент

CSS

- CSS — Cascading Style Sheets — каскадные таблицы стилей — это язык описания внешнего вида веб-страниц.
- Полный по Тьюрингу
- Работает в связке с HTML
- Большая экосистема

Основы HTML

Базовая структура: DOCTYPE

- Объявление типа документа
- Указывает браузеру, что документ написан на HTML5
- Всегда располагается первым в документе

```
<!DOCTYPE html>
```

Базовая структура: html

- Контейнер для контента
- Задаёт язык страницы

```
<html lang="ru">
```

Базовая структура: head

- Контейнер для метаинформации
- Можно задать название страницы и фавикон
- ... кодировку
- ... подключить стили
- ... добавить информацию для SEO

```
<head>
    <meta charset="UTF-8">
    <meta
        name="viewport"
        content="width=device-width, initial-scale=1.0"
    >
    <title>Заголовок страницы</title>
</head>
```

Базовая структура: body

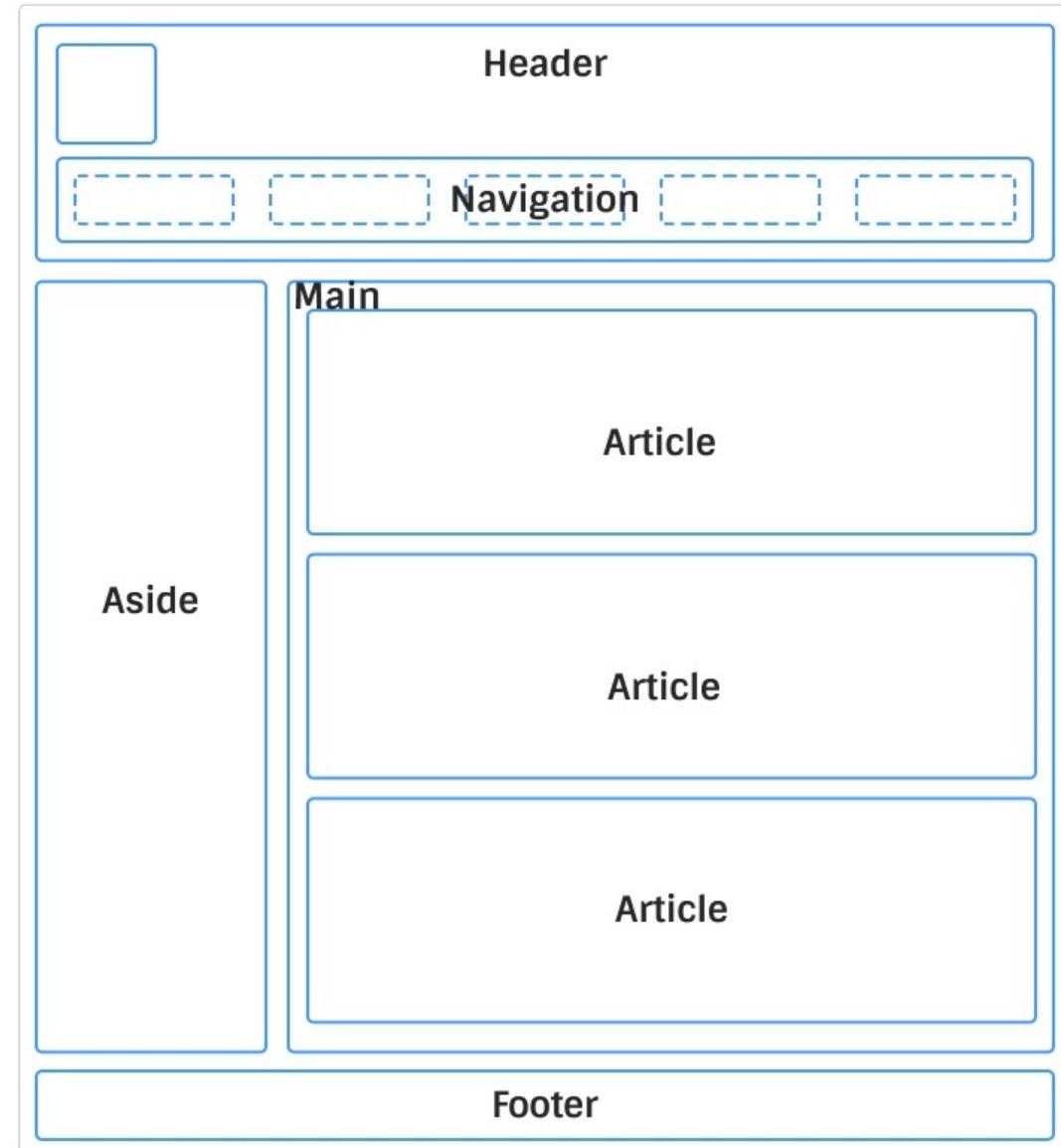
- Содержит весь видимый контент

```
<body>
    <!-- Весь видимый контент -->
</body>
```

Структурирующие теги

Структурирующие теги — это специальные элементы HTML5, которые помогают организовать контент веб-страницы в логические блоки.

- Семантичные
- Доступные
- Удобные
- SEO

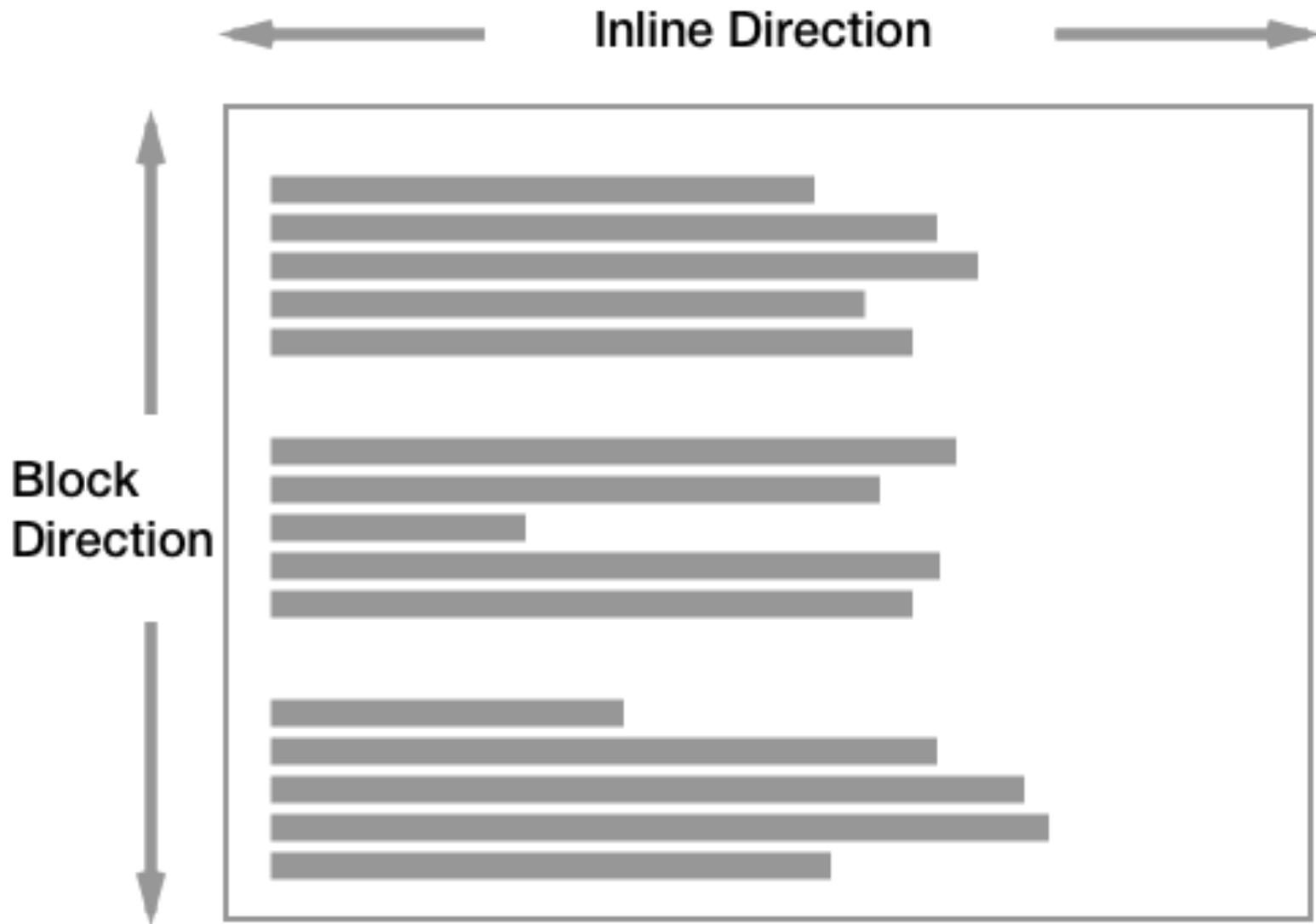


Остальные теги



<https://developer.mozilla.org/ru/docs/Web/HTML/Reference/Elements>

Поток документа



Поток документа



<https://doka.guide/html/flow/>

Основы CSS

Позиционирование

Позиционирование — это механизм в CSS, который позволяет управлять расположением элементов на веб-странице. Основной инструмент для этого — свойство position.

Позиционирование: static

- Значение по умолчанию
- Элемент находится в обычном потоке документа
- Не реагирует на свойства top, bottom, left, right

```
.element {  
    position: static;  
}
```

Позиционирование: relative

- Элемент смещается относительно своего исходного положения
- Занимает место в обычном потоке документа
- Можно использовать свойства смещения

```
.element {  
    position: relative;  
    top: 20px;  
    left: 15px;  
}
```

Позиционирование: absolute

- Выводится из обычного потока документа
- Позиционируется относительно ближайшего позиционированного родителя

```
.element {  
    position: absolute;  
    top: 40px;  
    left: 20px;  
}
```

Позиционирование: fixed

- Закрепляется относительно окна браузера
- Выводится из обычного потока

```
.element {  
    position: fixed;  
    top: 10px;  
    left: 10px;  
}
```

Позиционирование: sticky

- Ведёт себя как relative, пока не достигнет определённой позиции
- Затем фиксируется

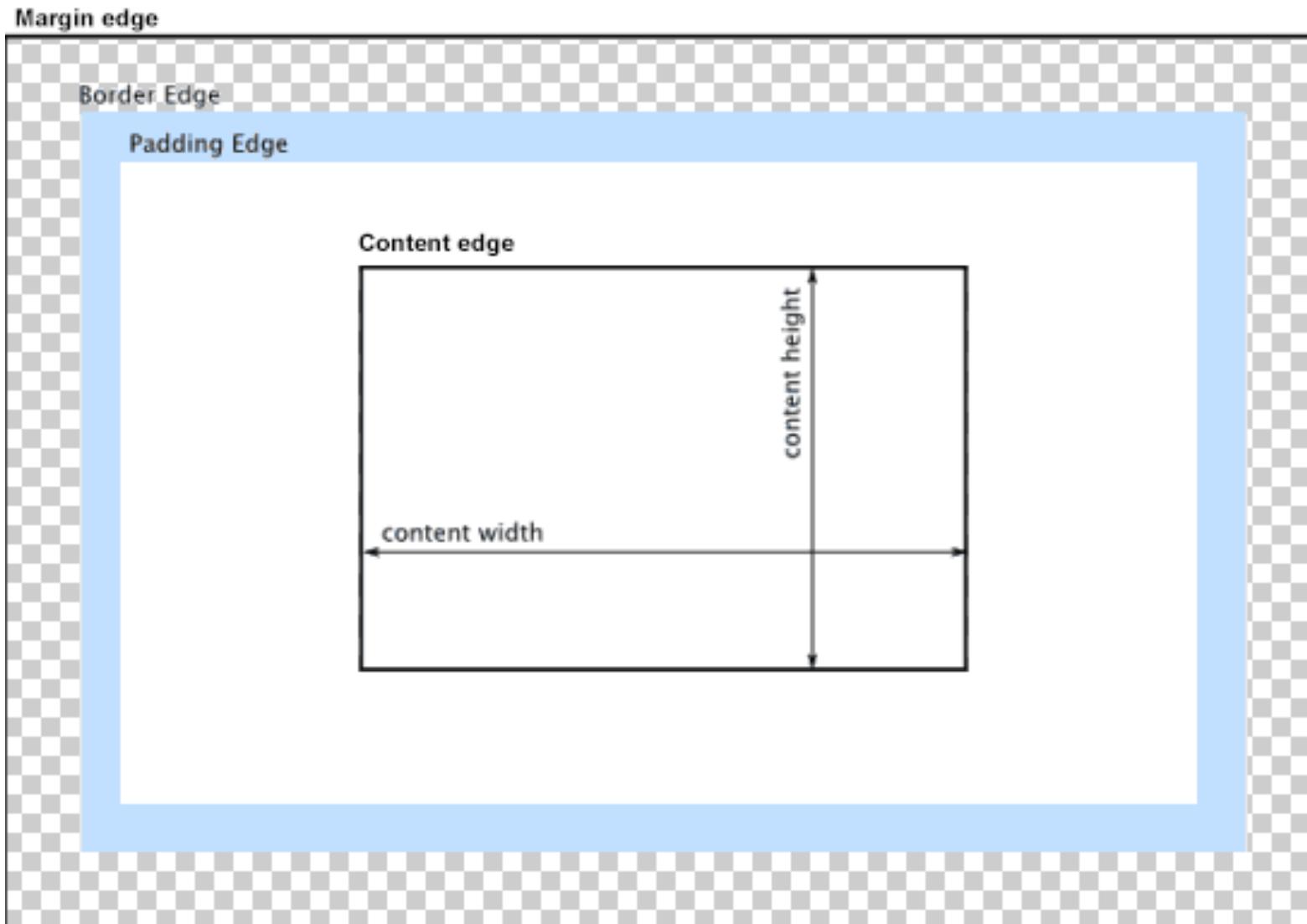
```
.element {  
    position: sticky;  
    top: 30px;  
}
```

Позиционирование: doka



<https://doka.guide/css/position/>

Блочная модель



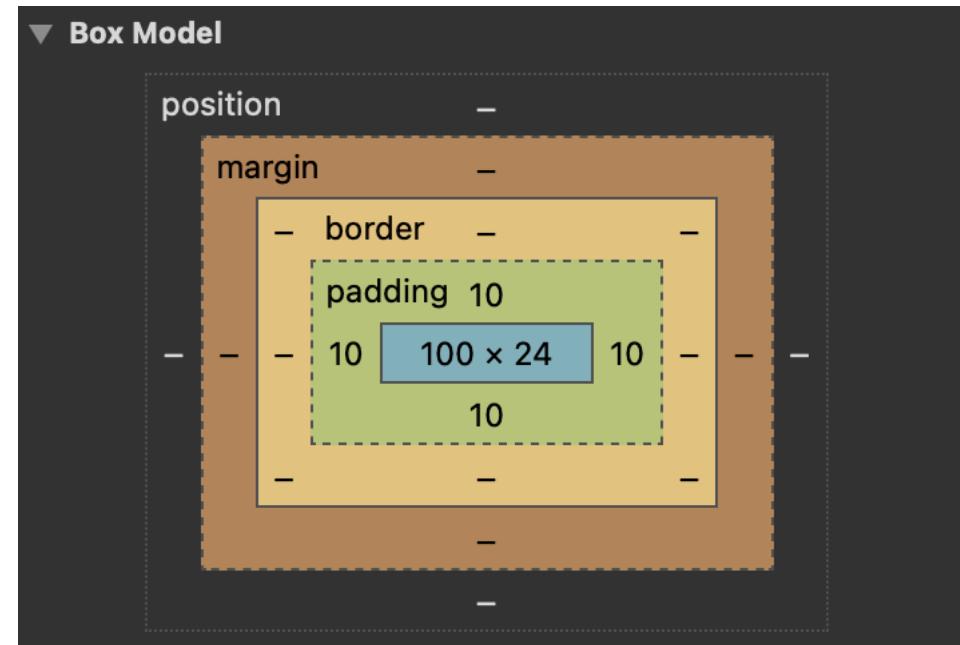
Блочная модель: box-sizing

Box-sizing — это CSS-свойство, которое изменяет способ вычисления ширины и высоты элементов в браузере. Оно позволяет контролировать, как будут учитываться границы (border) и внутренние отступы (padding) при определении размеров элемента.

Блочная модель: box-sizing = content-box

- Значение по умолчанию
- Ширина и высота включают только содержимое элемента
- Границы и внутренние отступы добавляются к указанным размерам

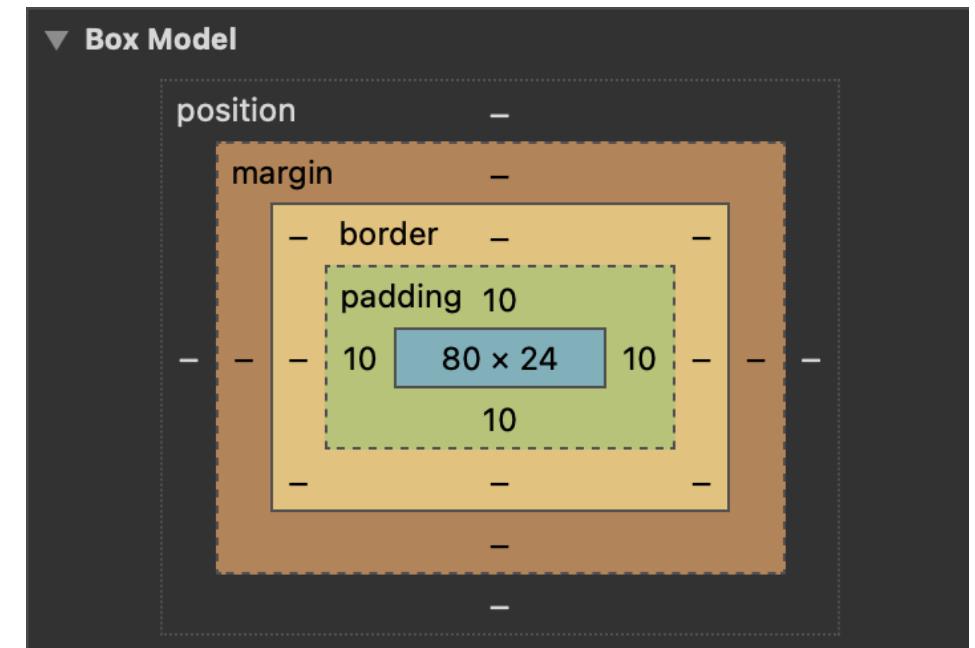
```
.element {  
    box-sizing: content-box;  
    width: 100px;  
    padding: 10px;  
}
```



Блочная модель: box-sizing = border-box

- Ширина и высота включают содержимое, внутренние отступы и границы
- Фактические размеры элемента остаются неизменными при добавлении padding и border

```
.element {  
    box-sizing: border-box;  
    width: 100px;  
    Padding: 10px;  
}
```



Блочная модель: doka



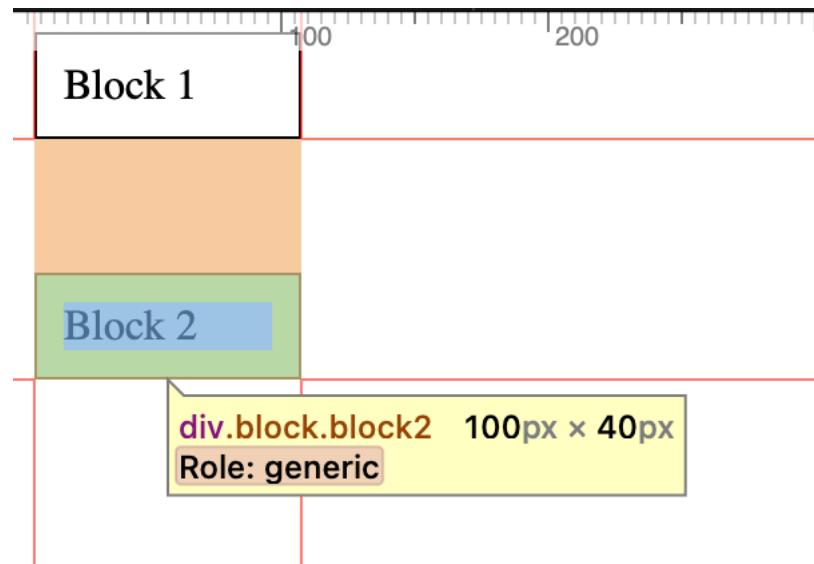
<https://doka.guide/css/box-model/>

Схлопывание отступов

Margin-collapse — это особенность CSS, при которой вертикальные внешние отступы соседних элементов не складываются, а «схлопываются», образуя единое пространство, равное большему из отступов.

```
<div class="block1"></div>
<div class="block2"></div>

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



Схлопывание отступов: mdn



https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_box_model/Mastering_margin_collapsing

Каскад

Каскад — это основной механизм в CSS, определяющий приоритет применения стилей, когда к одному элементу применяются несколько CSS-правил.

Что учитывается:

1. Специфичность селекторов
2. Важность (`!important`)
3. Источник
 1. Пользовательские настройки
 2. Inline-стили
 3. Внешние стили
 4. Стили браузера
4. Порядок объявления

Каскад: doka



<https://doka.guide/css/cascade/>

Специфичность

Специфичность — это механизм в CSS, который определяет приоритет применения стилей к элементу, когда к нему применяются несколько конфликтующих правил.

Порядок:

0. Inline-стили, !important -1000
1. Id - 100
2. Класс, атрибут, псевдокласс - 10
3. Тег, псевдоэлемент - 1

Специфичность: пример

Селектор	Вес
#block section > .list a	1.1.2
div#some	1.0.1
#some	1.0.0
* .list a	0.1.1
.class	0.1.0
section h1	0.0.2
section	0.0.1

Специфичность: doka



<https://doka.guide/css/specificity/>

Наследование

Наследование в CSS — это механизм передачи свойств от родительских элементов к их дочерним элементам.

Наследование: “семейные” свойства

- Шрифты: font-family, font-size, font-style, font-weight
- Цвет: color
- Текстовые параметры: line-height, letter-spacing, word-spacing, white-space, text-align, vertical-align

Наследование: “индивидуальные” свойства

- Отступы и границы: margin, padding, border
- Оформление: background, box-shadow
- Позиционирование: position, float, clear
- Свойства блока: display, width, height, overflow, z-index

Наследование: управление

- `inherit` - принудительное наследование любого свойства
- `initial` - принудительный сброс в значение по умолчанию (может быть неожиданным)

Наследование: doka



<https://doka.guide/css/inheritance/>

Свойство `display`

Display — это ключевое свойство CSS, которое определяет, как элемент будет отображаться на веб-странице и как он будет взаимодействовать с другими элементами.

Не анимируется (пока)

Display: block

- Занимает всю доступную ширину
- Начинается с новой строки

```
.element {  
    display: block;  
}
```

Display: inline

- Не создает новую строку
- Занимает только необходимое пространство
- Не реагирует на width/height

```
.element {  
    display: inline;  
}
```

Display: inline-block

- Снаружи inline, внутри block
- Можно задавать размеры
- Располагается в строке

```
.element {  
    display: inline-block;  
}
```

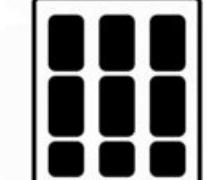
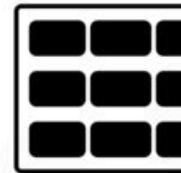
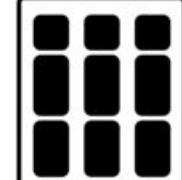
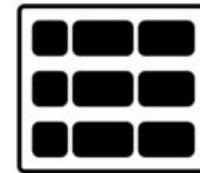
Display: flex

- Создает flex-контейнер
- Снаружи block (есть inline-flex)
- Позволяет гибко управлять расположением элементов

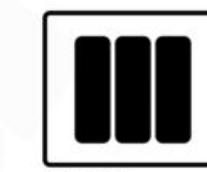
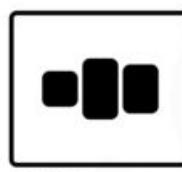
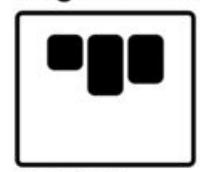
```
.element {  
    display: flex;  
}
```

CSS Flexbox

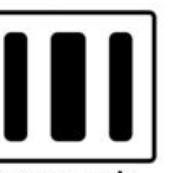
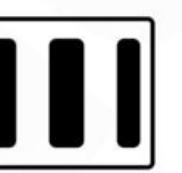
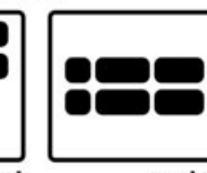
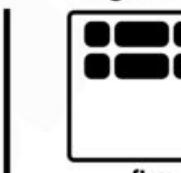
flex-direction



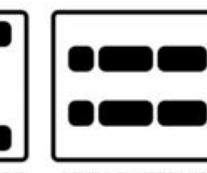
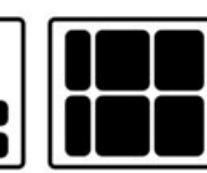
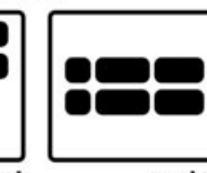
align-items



justify-content

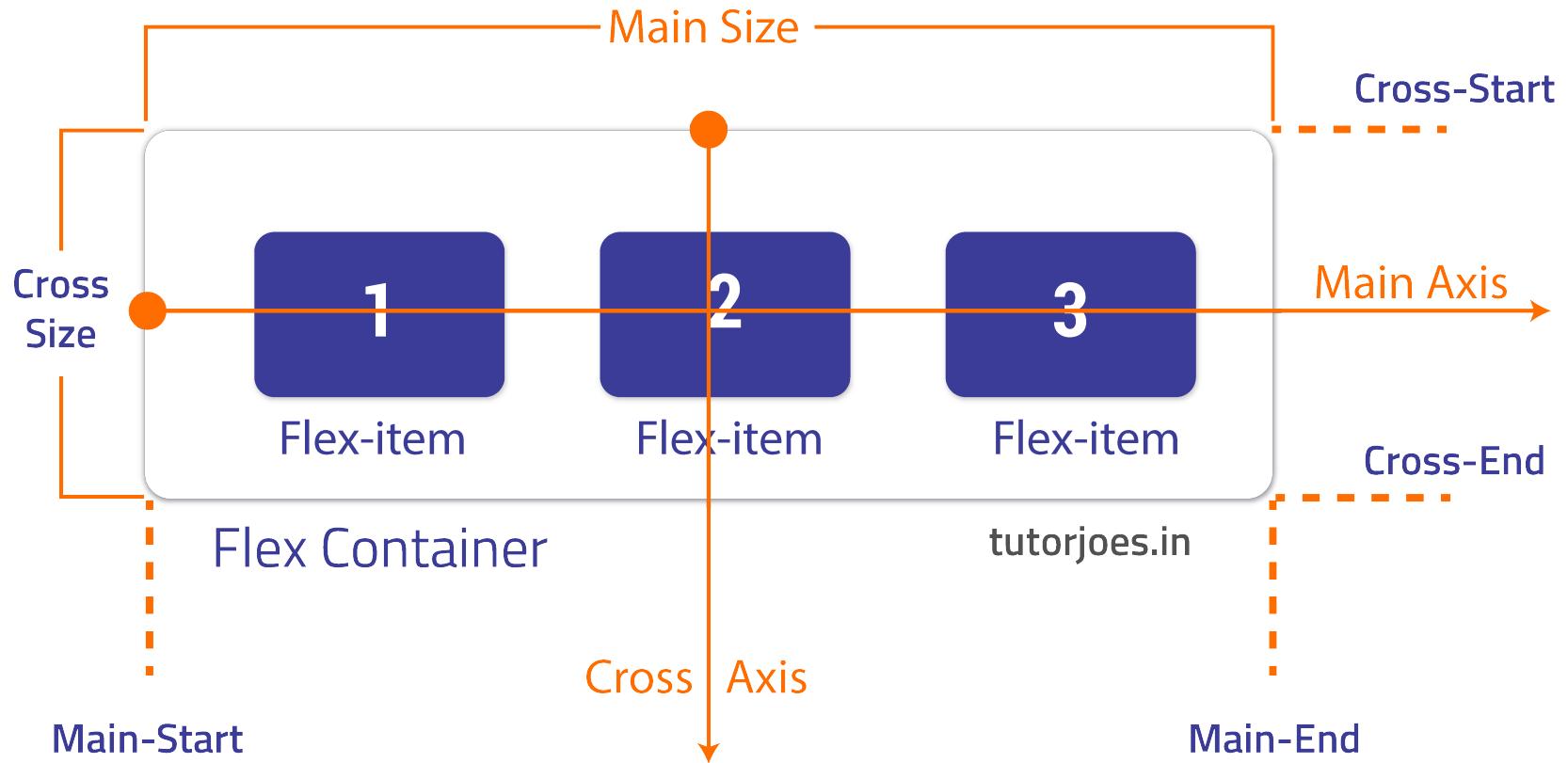


align-content



Display: flex

FlexBox Architecture



Display: flex: примеры



<https://yoksel.github.io/flex-cheatsheet/>

Display: flex: doka



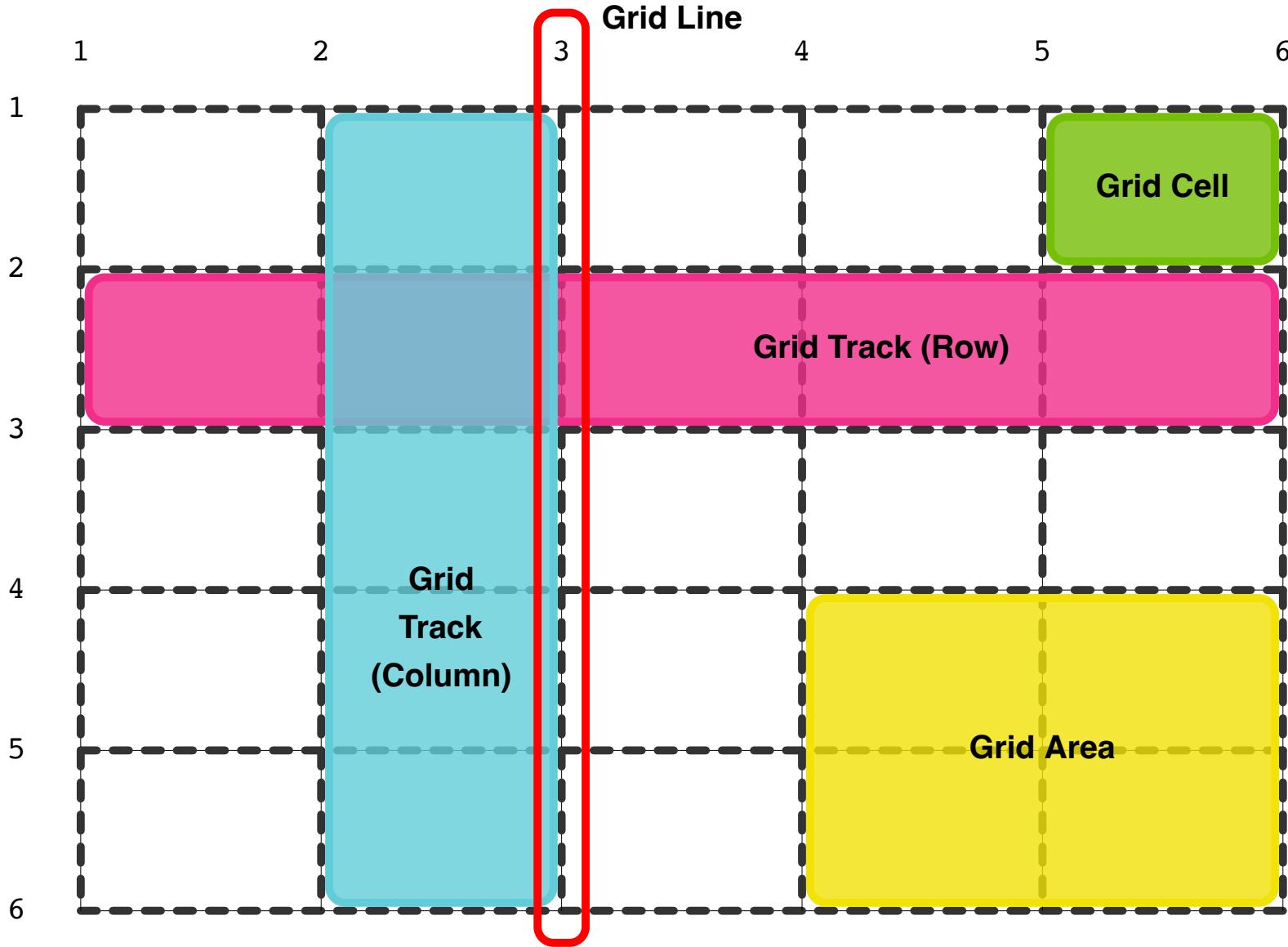
<https://doka.guide/css/flexbox-guide/>

Display: grid

- Создает grid-контейнер
- Позволяет создавать сложные макеты

```
.element {  
    display: grid;  
}
```

Display: grid



Display: grid: doka



<https://doka.guide/css/grid-guide/>

Display: none

- Полностью убирает элемент из потока документа
- Не занимает места

```
.element {  
    display: none;  
}
```

Display: table

- Позволяет строить таблицы как будто с помощью тега table, но без него
- Редко используется

```
.element {  
    display: table;  
}
```

Display: doka



<https://doka.guide/css/display/>

DevTools

DevTools

DevTools — это встроенный в браузер набор инструментов для разработки, отладки и тестирования веб-приложений. Они позволяют разработчикам анализировать и изменять содержимое веб-страниц в реальном времени.

TailwindCSS

Tailwind

Tailwind CSS — это современный CSS-фреймворк, основанный на утилитарном подходе к стилизации веб-страниц. В отличие от традиционных фреймворков, Tailwind не предоставляет готовых компонентов, а предлагает набор низкоуровневых утилитных классов для создания пользовательских интерфейсов.

Tailwind

Tailwind CSS — это современный CSS-фреймворк, основанный на утилитарном подходе к стилизации веб-страниц. В отличие от традиционных фреймворков, Tailwind не предоставляет готовых компонентов, а предлагает набор низкоуровневых утилитных классов для создания пользовательских интерфейсов.

Tailwind: утилиты

- Макет (flex, grid, float)
- Пространственные (margin, padding)
- Типографика (font-size, line-height)
- Цвета (text-color, bg-color)
- Эффекты (shadow, opacity)
- Анимации (transition, transform)

Tailwind: пример

```
<div class="bg-blue-500 p-4 rounded shadow-md">
  <h1 class="text-2xl font-bold">Заголовок</h1>
  <p class="text-gray-700">Текст параграфа</p>
</div>
```

Tailwind: ссылка



<https://tailwindcss.com/>

SEO

SEO

SEO (Search Engine Optimization) — это комплекс мер по оптимизации веб-сайта для повышения его позиций в результатах поисковых систем по определённым запросам пользователей.

SEO: основные компоненты

- Техническая оптимизация: скорость загрузки сайта, мобильная адаптивность, структура URL, карта сайта (sitemap), файл robots.txt
- Внутренняя оптимизация: оптимизация контента, мета-теги, внутренние ссылки, структурированные данные
- Внешняя оптимизация: построение ссылочного профиля, социальные сигналы, репутация домена

SEO: важный HTML

- title: уникальное название страницы
- meta: необходимая мета-информация
- h1-h6: структурирование контента
- семантические теги (header, footer, nav, article)
- Атрибут alt у изображений
- a: ссылки

SEO: валидаторы

- Google Search Console
- Яндекс.Вебмастер
- Валидаторы HTML
- Анализаторы производительности

SEO: doka



<https://doka.guide/html/seo-for-beginners/>

Адаптивность

Адаптивность

Адаптивный веб-дизайн — это подход к разработке веб-сайтов, который позволяет автоматически подстраивать интерфейс под различные устройства, размеры экрана и ориентации.

Адаптивность: принципы

Гибкая сетка

- Использование процентов вместо пикселей
- Применение медиа-запросов
- Адаптивные изображения

Адаптивность: принципы

Отзывчивый макет

- Перестраивание элементов при изменении размера окна
- Оптимизация расположения блоков
- Изменение размеров шрифтов

Адаптивность: принципы

Оптимизация контента

- Скрытие второстепенных элементов
- Адаптация изображений
- Изменение структуры меню

Адаптивность: проверки

- Google Mobile-Friendly Test
- PageSpeed Insights
- BrowserStack
- Responsive Design Checker

Вопросы