



# ОСНОВЫ WEB-ТЕХНОЛОГИЙ

Лекция 3

# СТРУКТУРА ЗАНЯТИЯ

- CSS внешний вид сайта
- Работа с цветами
- Мастер класс по работе с макетом
- Что такое шаблонизация и зачем нужна

# CASCADING STYLE SHEETS(CSS)

CSS (англ. **Cascading Style Sheets** — каскадные таблицы стилей) — формальный язык описания внешнего вида документа, который написан с использованием языка разметки HTML.

Подготовкой и выпуском спецификации занимается консорциум W3C -

<http://www.w3.org/Style/CSS/>

# CASCADING STYLE SHEETS(CSS)

Исторически CSS появился вместе с HTML4.01(**17 декабря 1996**) для упрощения манипулирования разметкой, поскольку именно использование стилей позволяет осуществлять групповую замену шрифта, цвета, размера и взаимного расположения элементов.



Концепция *каскадных таблиц стилей* была предложена Хоконом Виум Ли

# К ПРЕИМУЩЕСТВАМ ИСПОЛЬЗОВАНИЯ CSS ОТНОСЯТСЯ:

1. *Централизованное управление отображением множества документов при помощи одной таблицы стилей;*
2. Упрощенный контроль внешнего вида веб - страниц;
3. Наличие разработанных дизайнерских техник;
4. Возможность использования различных стилей для одного документа, в зависимости от устройства, при помощи которого осуществляется доступ к веб - странице.

# ОБЪЕКТНАЯ МОДЕЛЬ БРАУЗЕРА

Стандарт *HTML 4.0* помимо *CSS* утвердил и объектную модель браузера (*Browser Object Model – BOM*)

**Объектная модель браузера** описывает содержимое веб - документа, т.е. сама модель является набором объектов, описывающих указанное содержимое. Поскольку *BOM* уникальна для каждого браузера, возникали проблемы с межплатформенными приложениями.

В конечном итоге, на место объектной модели браузера пришла **объектная модель документа (*Document Object Model – DOM*)**, описывающая стандарт представления веб-страниц в виде набора объектов.

# СИНТАКСИС ОПИСАНИЯ СТИЛЯ CSS

селектор { свойство: значение; }



описание

правило

# ОСНОВНЫЕ ПОНЯТИЯ

**Селектор (Selector).** Селектор — это элемент, к которому будут применяться назначаемые стили. Это может быть тег, класс или идентификатор объекта гипертекстового документа.

**Свойство (Property).** Свойство определяет одну или несколько характеристик селектора. Свойства задают формат отображения селектора: отступы, шрифты, выравнивание, размеры и т.д.

**Значение (Value).** Значения — это фактические числовые или строковые константы, определяющие свойство селектора.

**Описание (Declaration).** Совокупность свойств и их значений.

**Правило (Rule).** Полное описание стиля (селектор + описание)



# ВАРИАНТЫ ПОДКЛЮЧЕНИЯ CSS

**Встраивание (Inline).** Этот метод позволяет применить стиль к заданному тегу HTML.

**Внедрение (Embedded).** Внедрение позволяет управлять стилями страницы целиком.

**Связывание (Linked или External).** Связанная таблица стилей позволяет вынести описание стилей во внешний файл, ссылаясь на который можно контролировать отображение всех страниц сайта.

# ПОДКЛЮЧЕНИЕ ФАЙЛА CSS

**<link rel="stylesheet" href="mysite.css">**

**<link rel="stylesheet"**  
**href="http://www.htmlbook.ru/main.css">**

# ПОРЯДОК ПРИМЕНЕНИЯ СТИЛЕЙ

1. Если правила одинаковые, то будет применён самый последний
2. Чем конкретнее написан селектор, тем приоритетнее будет стиль
3. Инлайн стиль приоритетнее всего
4. Чит: !important

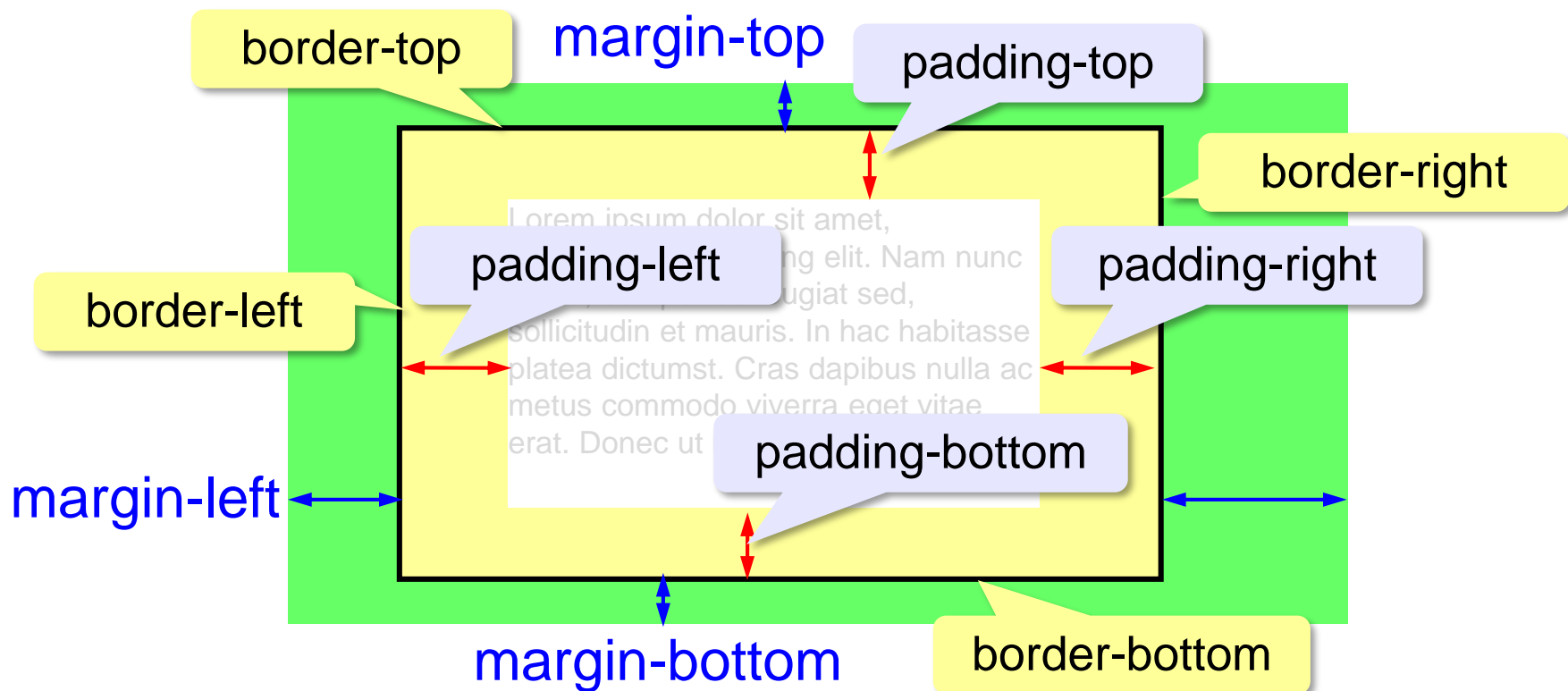
# СЕЛЕКТОРЫ CSS

1. По имени тега
2. По атрибуту class
3. По атрибуту id

Другие типы селекторов:

30 CSS-селекторов, о которых  
полезно помнить  
([codeharmony.ru](http://codeharmony.ru))

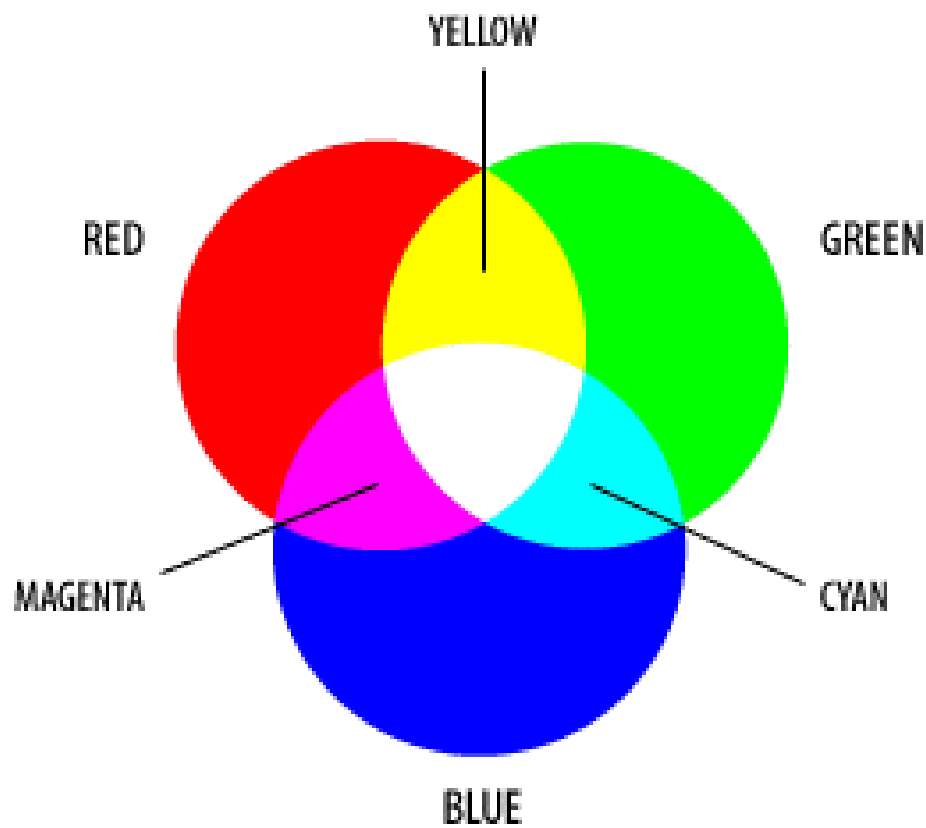
# БЛОЧНАЯ МОДЕЛЬ



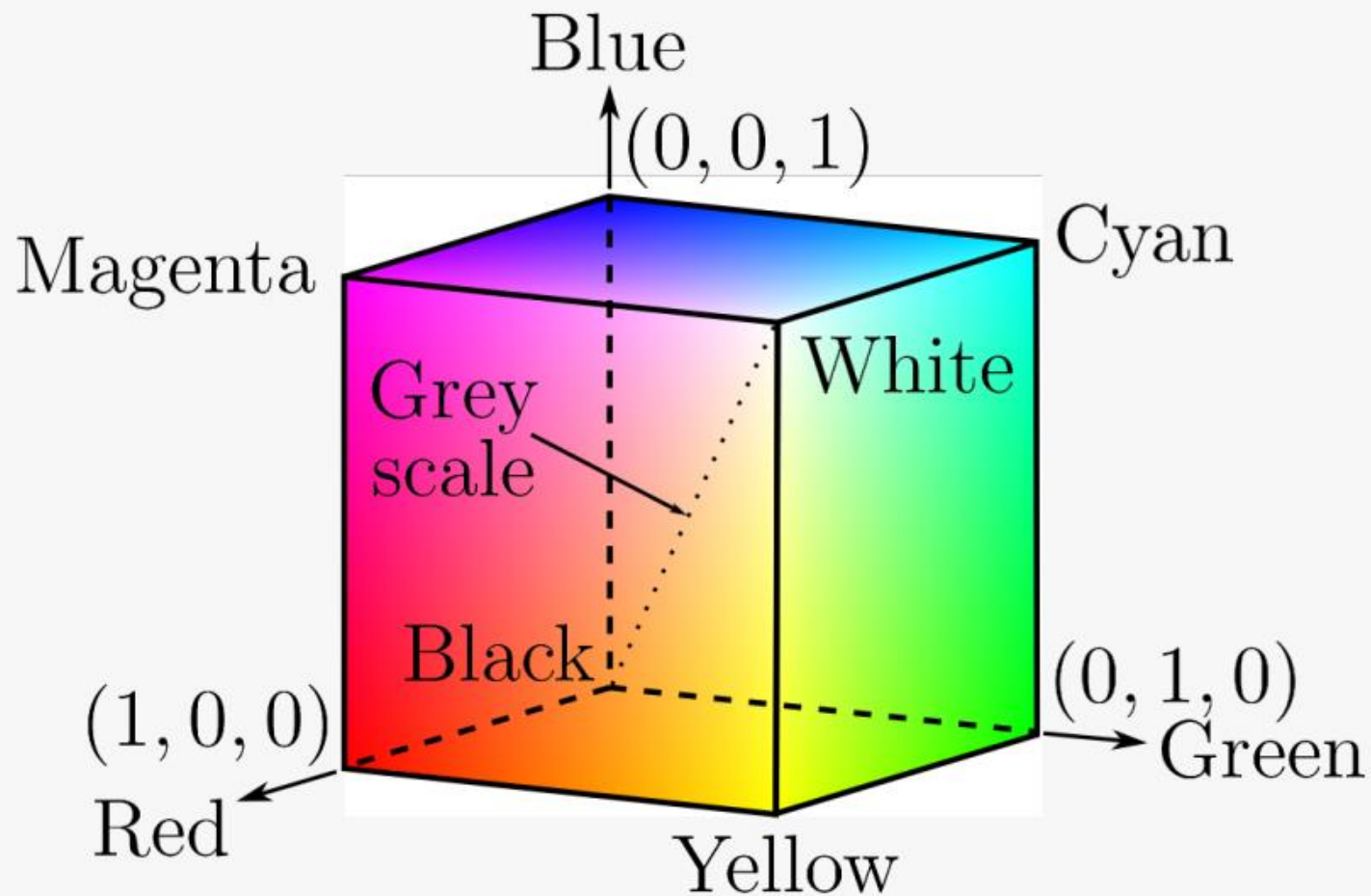
# CSS ЕДИНИЦЫ ИЗМЕРЕНИЯ

- px
- em
- %
- rem
- Vh, vw

# ВЕБ-ЦВЕТА. МОДЕЛЬ RGB

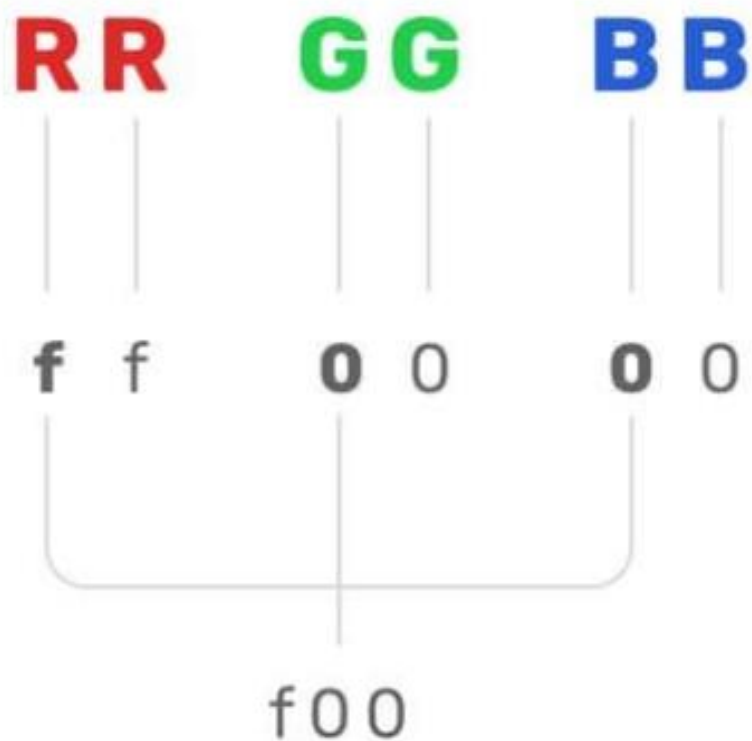


# МОДЕЛЬ RGB





# ШЕСТНАДЦАТЕРИЧНЫЕ ЦВЕТОВЫЕ ЗНАЧЕНИЯ



# ЦВЕТА, ЗАДАВАЕМЫЕ В ФОРМАТЕ RGB/RGBA

```
.element {  
    /* Белый */  
    color: rgb(255, 255, 255);  
    color: rgb(100%, 100%, 100%);  
  
    /* Чёрный */  
    background-color: rgb(0, 0, 0);  
    background-color: rgb(0%, 0%, 0%);  
}
```

РАСШИРЕНИЕ ДЛЯ БРАУЗЕРА

ColorZilla

# CSS ПОЗИЦИОНИРОВАНИЕ

- position static, fixed, absolute, relative
- float: left

# ЧТО ТАКОЕ FLEXBOX?

**Флексбокс** — это CSS-механизм, который позволяет контролировать размер, порядок и выравнивание элементов по нескольким осям, распределение свободного места между элементами и многое другое.

Чтобы включить флексбокс, нужно задать элементу свойство `display: flex;`. После этого:

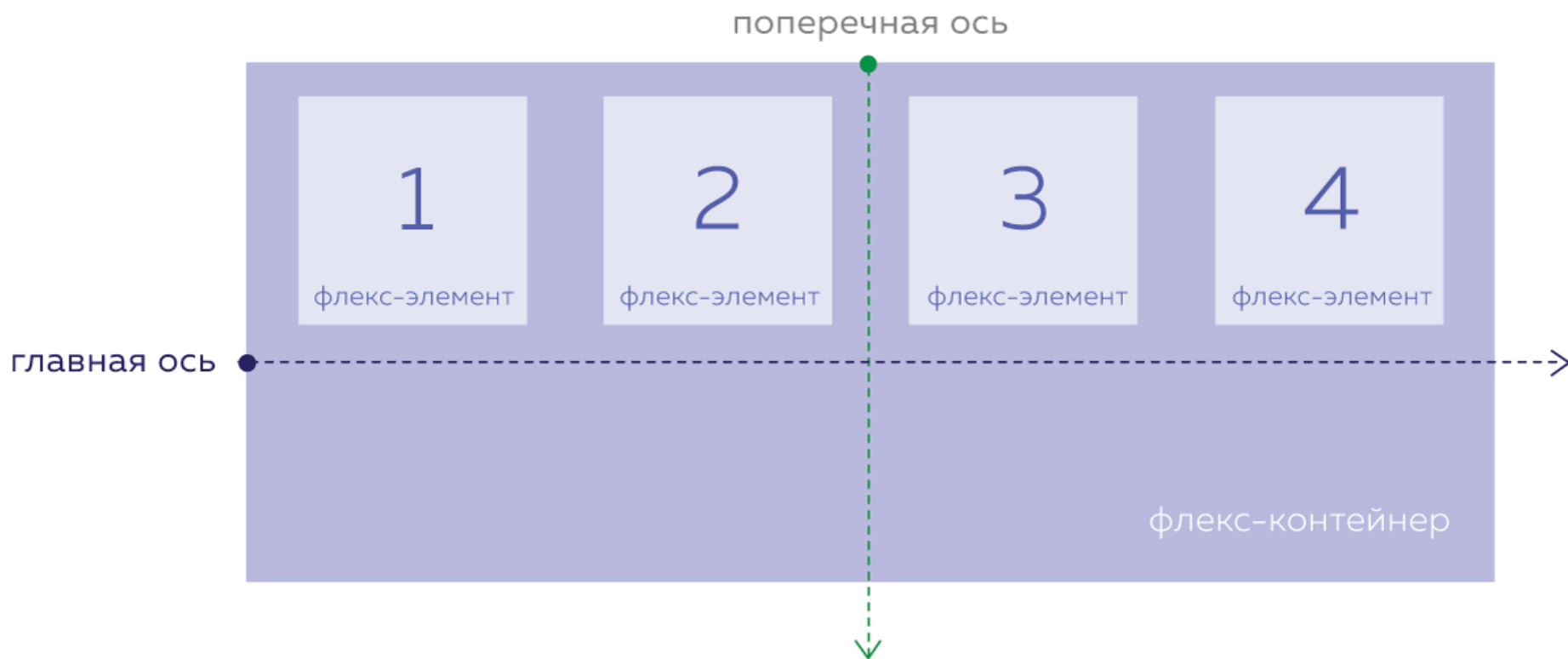
1. Элемент с `display: flex;` превращается во «флекс-контейнер».
2. Непосредственные потомки этого элемента превращаются во «флекс-элементы» и начинают распределяться по новым правилам.

# ГЛАВНАЯ И ПОПЕРЕЧНАЯ ОСИ

**Главной осью** flex-контейнера является направление, в соответствии с которым располагаются все его дочерние элементы. Поток флекс-элементов «течёт» вдоль главной оси от её начала к её концу.

**Поперечной осью** называется направление, перпендикулярное главной оси. Вдоль этой оси работают «вертикальные» выравнивания.

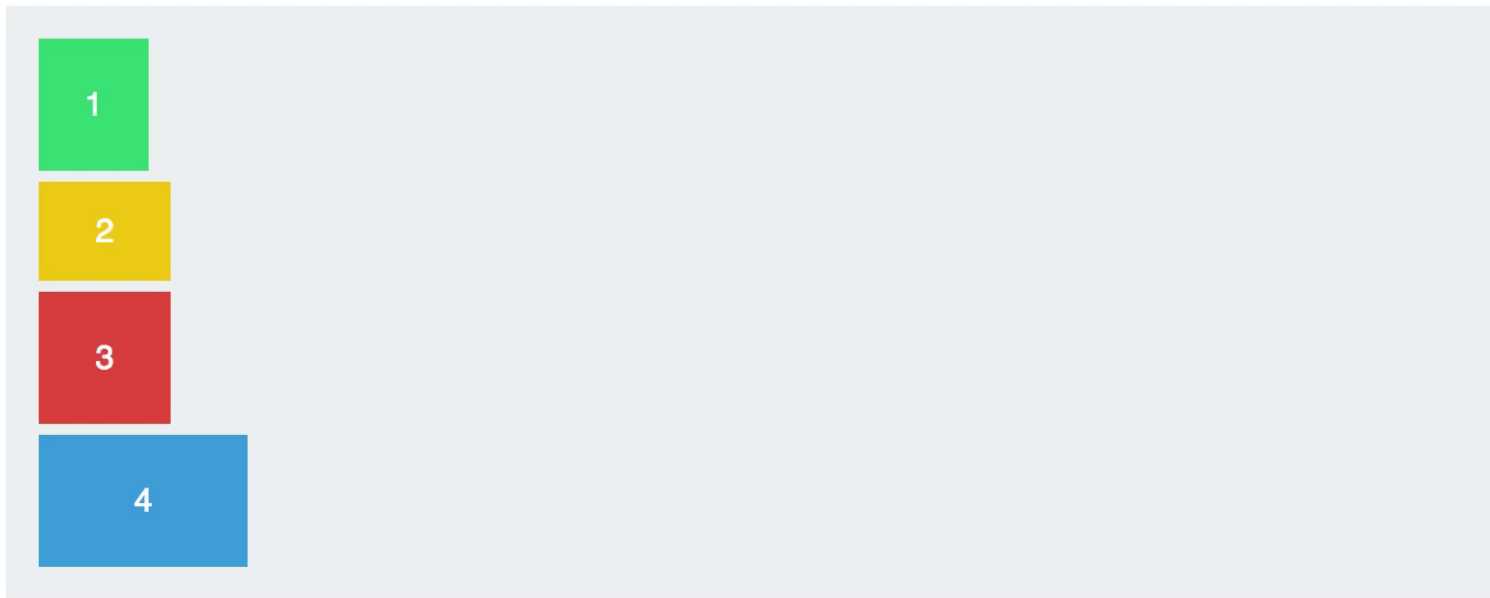
# ГЛАВНАЯ И ПОПЕРЕЧНАЯ ОСИ



# КАК ЭТО УСТРОЕНО?

Пример

**display: block;**





# ГЛАВНАЯ ОСЬ

По умолчанию главная ось направлена слева направо, но её можно разворачивать во всех направлениях с помощью свойства `flex-direction`, которое задаётся для флекс-контейнера. Значения свойства:

- Значение по умолчанию **row** — главная ось направлена слева направо.
- **column** — главная ось направлена сверху вниз.
- **row-reverse** — главная ось направлена справа налево.
- **column-reverse** — главная ось направлена снизу вверх.

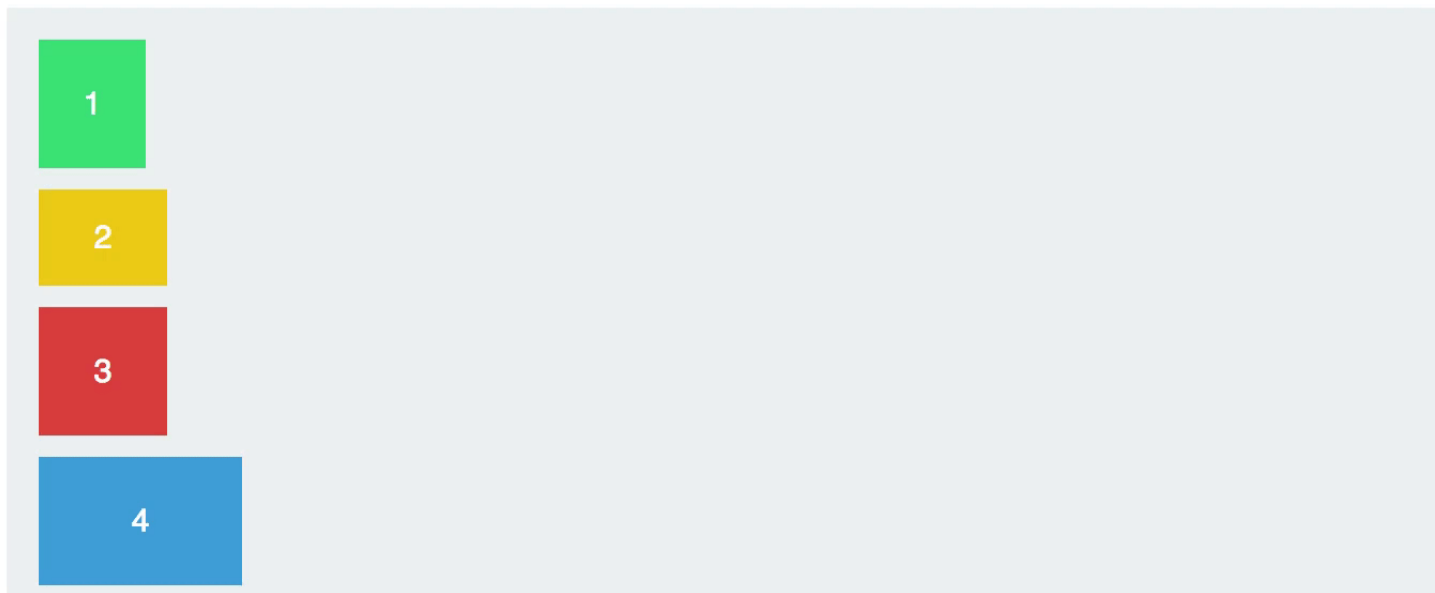
# КАК ЭТО УСТРОЕНО?

**flex-direction: row;**



# КАК ЭТО УСТРОЕНО?

`flex-direction: column;`



# ПОПЕРЕЧНАЯ ОСЬ

Поперечная ось всегда перпендикулярна главной оси и поворачивается вместе с ней:

- Если главная ось направлена горизонтально, то поперечная ось смотрит вниз.
- Если главная ось направлена вертикально, то поперечная ось смотрит направо.

Таким образом, поперечная ось никогда не смотрит вверх или влево, и свойства для поворота поперечной оси нет.

# РАСПРЕДЕЛЕНИЕ ФЛЕКС-ЭЛЕМЕНТОВ

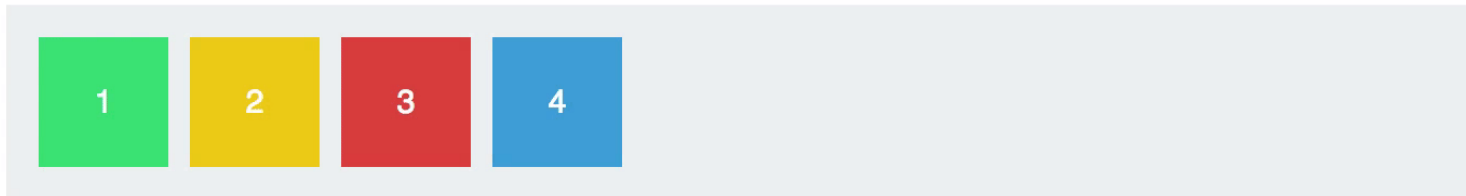
## *Выравнивание по главной оси*

CSS-свойство **justify-content** определяет то, как будут выровнены элементы вдоль главной оси. Доступные значения justify-content:

- Значение по умолчанию **flex-start** — элементы располагаются у начала главной оси.
- **flex-end** — элементы располагаются в конце главной оси.
- **center** — элементы располагаются по центру главной оси.
- **space-between** — элементы располагаются так, что расстояния между соседними одинаковые, а между элементами и краями флекс-контейнера отступов нет.
- **space-around** — элементы располагаются так, что расстояния между соседними одинаковые, а между элементами и краями флекс-контейнера есть отступ, равный половине расстояния между соседними элементами.
- **space-evenly** — расстояния между соседними элементами и краями флекс-контейнера одинаковые.

# КАК ЭТО РАБОТАЕТ?

**`justify-content: flex-start;`**



# РАСПРЕДЕЛЕНИЕ ФЛЕКС-ЭЛЕМЕНТОВ

## *Выравнивание по поперечной оси*

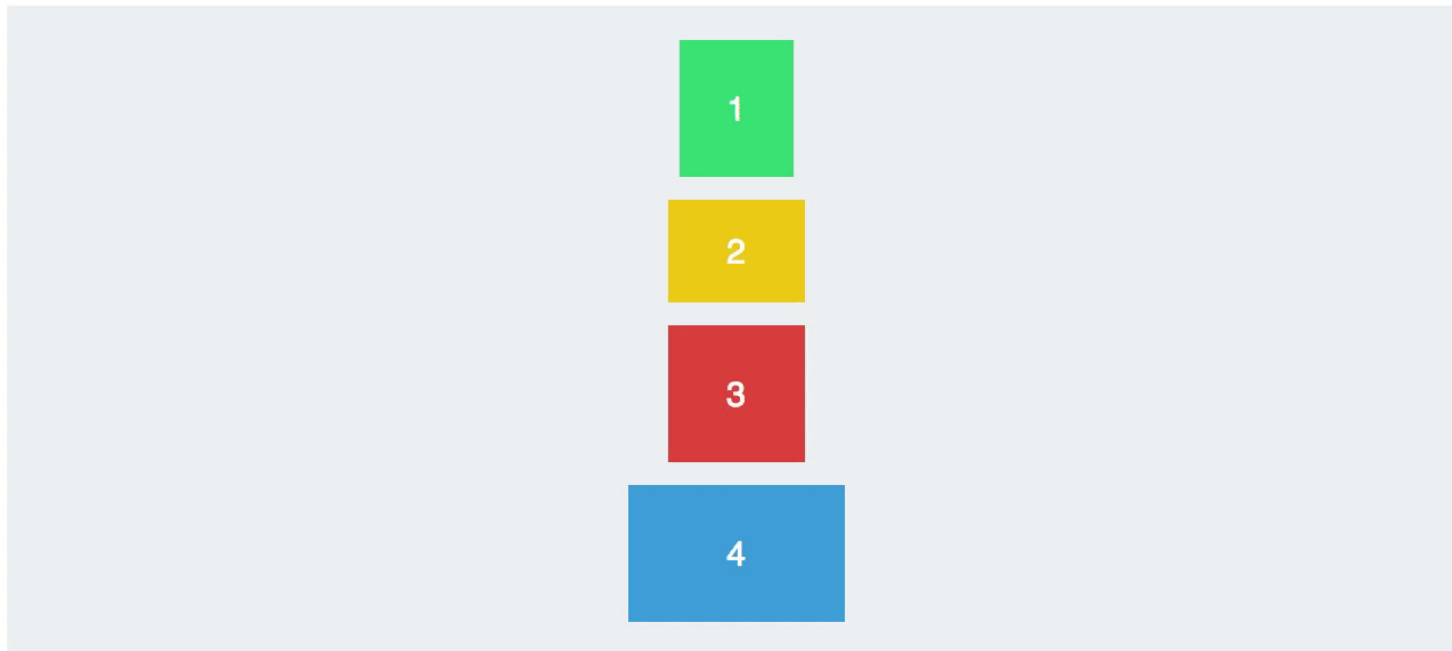
CSS-свойство **align-items** определяет то, как будут выровнены элементы вдоль поперечной оси. Доступные значения align-items:

- Значение по умолчанию **stretch** — элементы растягиваются на всю «высоту» флекс-контейнера.
- **flex-start** — элементы располагаются у начала поперечной оси.
- **flex-end** — элементы располагаются в конце поперечной оси.
- **center** — элементы располагаются по центру поперечной оси.
- **baseline** — элементы выравниваются по базовой линии текста внутри них.

Распределение элементов по главной оси задаётся для всего флекс-контейнера и на все флекс-элементы действует одинаково, задать какому-то элементу отличное от других распределение по главной оси нельзя.

# КАК ЭТО УСТРОЕНО?

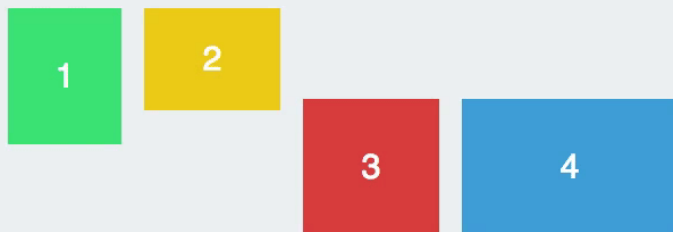
**flex-direction: column;**  
**justify-content: center; align-items: center;**





# КАК ЭТО УСТРОЕНО?

**align-self: flex-start;**



# ПОРЯДКОВЫЙ НОМЕР ФЛЕКС-ЭЛЕМЕНТА

Порядок следования флекс-элементов в потоке можно изменять с помощью свойства `order`, порядкового номера флекс-элемента, не меняя при этом HTML-код.

По умолчанию порядковый номер флекс-элементов равен 0, а сортировка элементов производится по возрастанию номера. Порядковый номер задаётся целым числом, положительным или отрицательным.

# CSS НАСЛЕДОВАНИЕ

- color:
- font-size:
- letter-spacing:



# СОДЕРЖИМОЕ САЙТА

index.html

# СОДЕРЖИМОЕ САЙТА

index.html

style.css

script.js

# СОДЕРЖИМОЕ САЙТА

index.html

main.css

main.js

menu.css

menu.js

media.css

gallery.js

home.css

bootstrap.js

simplePage.js

# СОДЕРЖИМОЕ САЙТА

index.html

main.css

main.js

logo.svg

menu.css

menu.js

hello.jpg

media.css

gallery.js

image.png

home.css

bootstrap.js

video.mp4

simplePage.js

audio.mp3

и другие...

# МЕДИА-ЗАПРОСЫ CSS

**Медиа-запросы** — это особые правила применения CSS, с помощью которых можно менять стиль элементов, отталкиваясь от устройств, на которых будут просматривать веб-страницу.

@media условие { /\* стили, которые становятся активными при соответствии условию }



# ВИДЫ ЗАПРОСОВ

все устройства — all;

устройства с экранами — screen;

синтезаторы речи — speech;

принтеры или страница

предварительного просмотра для  
печати — print

.

# ВИДЫ УСЛОВИЙ

**Height, min-height и max-height** — прописывает требования к высоте браузера, аналогично функциям ширины: @media (max-height: 1080px)

**Color** определяет количество бит на канал цвета (также есть значения min и max);

**aspect-ratio (min-aspect-ratio, max-aspect-ratio)** — отношения ширины к высоте: @media screen and (device-aspect-ratio: 1920/1080)

**orientation** — оценивает режим отображения страницы: портретный (portrait) или альбомный (landscape);

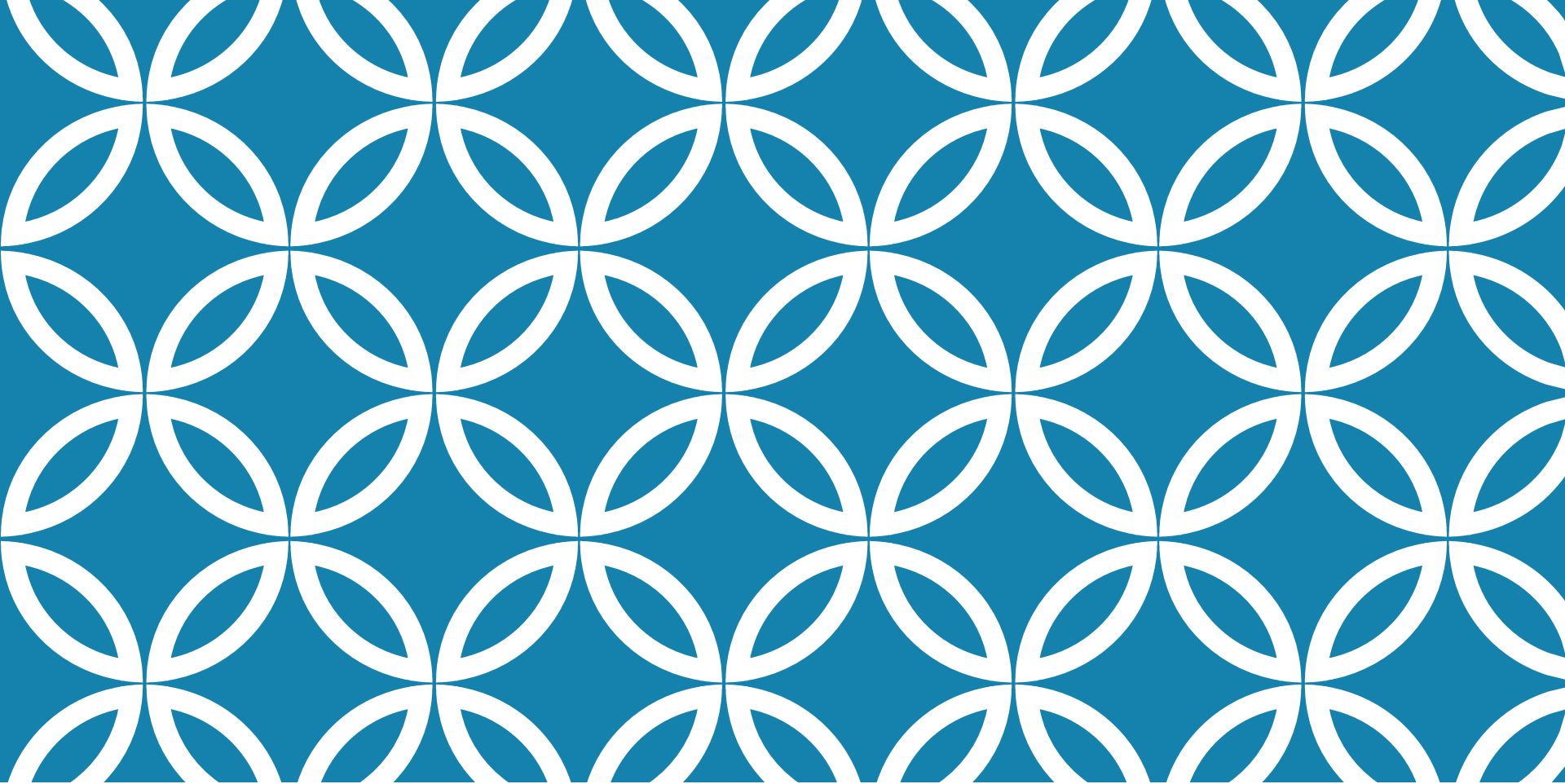
**Resolution (min-resolution, max-resolution)** — это разрешение устройства ввода: требуемое, минимальное и максимальное; измеряется в dpi, dpcm или dppx.

# ПРОБЛЕМЫ С БРАУЗЕРАМИ

Обязательно просматривайте страницы с таблицами стилей в различных браузерах. Это связано с тем, что разные браузеры могут по-разному интерпретировать одно и то же правило, а некоторые свойства и/или значения и вовсе не поддерживать.

# ПОПРОБОВАТЬ СВЕРСТАТЬ САЙТ САМИМ

[https://www.figma.com/file/riYWskz8Mltxiv4D6nN6lo/Free--Landing--Page-Template-\(Copy\)?node-id=254%3A515](https://www.figma.com/file/riYWskz8Mltxiv4D6nN6lo/Free--Landing--Page-Template-(Copy)?node-id=254%3A515)



**СПАСИБО ЗА  
ВНИМАНИЕ!**

конец лекции 2