

A decorative graphic on the left side of the slide. It consists of a blue parallelogram and a light green parallelogram, both tilted at an angle. The blue shape is in the foreground, and the green shape is partially behind it. They are set against a dark blue background with faint, lighter blue diagonal stripes.

CSS, Сетки, БЭМ



Темы раздела

- ❖ Введение в CSS
- ❖ Подключение шрифтов
- ❖ Сетки
- ❖ Flexbox
- ❖ БЭМ



Введение в CSS



Синтаксис CSS

```
селектор {  
    СВОЙСТВО: значение;  
    СВОЙСТВО: значение;  
    СВОЙСТВО: значение;  
}  
селектор2 {  
    СВОЙСТВО: значение;  
    СВОЙСТВО: значение;  
}
```

```
/* Комментарий пишется таким образом */
```



Селекторы

```
p {...}  
.popular-item {...}  
p.title {...}  
#banner {...}  
h2#title {...}  
.popular-item img  
  
.popular .popular-item {...}
```

```
/* по тегу */  
/* по классу */  
/* тег с классом */  
/* по id */  
/* тег с id */  
/* тег внутри тега с  
классом */  
/* класс внутри тега с  
классом*/
```



p {...}

```
<!DOCTYPE html>
<html lang="ru">
  <head>
    <meta charset="utf-8">
    <title>Заголовок вкладки браузера</title>
  </head>
  <body>
    <ul class="site-navigation">
      <li class="popular"><p class="popular-item">Информация</p></li>
      <li><p class="title">Новости</p></li>
      <li id="banner"><h2 id="title">Новости</h2></li>
      <li><p class="popular-item"></p></li>
    </ul>
  </body>
</html>
```

.popular-item {...}

```
<!DOCTYPE html>
<html lang="ru">
  <head>
    <meta charset="utf-8">
    <title>Заголовок вкладки браузера</title>
  </head>
  <body>
    <ul class="site-navigation">
      <li class="popular"><p class="popular__item">Информация</p></li>
      <li><p class="title">Новости</p></li>
      <li id="banner"><h2 id="title">Новости</h2></li>
      <li><p class="popular__item"></p></li>
    </ul>
  </body>
</html>
```

p.title {...}

```
<!DOCTYPE html>
<html lang="ru">
  <head>
    <meta charset="utf-8">
    <title>Заголовок вкладки браузера</title>
  </head>
  <body>
    <ul class="site-navigation">
      < li class="popular"><p class="popular__item">Информация</p></li>
        <li><p class="title">Новости</p></li>
      <li id="banner"><h2 id="title">Новости</h2></li>
      <li><p class="popular__item"></p></li>
    </ul>
  </body>
</html>
```


#banner {...}

```
<!DOCTYPE html>
<html lang="ru">
  <head>
    <meta charset="utf-8">
    <title>Заголовок вкладки браузера</title>
  </head>
  <body>
    <ul class="site-navigation">
      < li class="popular"><p class="popular__item">Информация</p></li>
      <li><p class="title">Новости</p></li>
      <li id="banner"><h2 id="title">Новости</h2></li>
      <li><p class="popular__item"></p></li>
    </ul>
  </body>
</html>
```

h2#title {...}

```
<!DOCTYPE html>
<html lang="ru">
  <head>
    <meta charset="utf-8">
    <title>Заголовок вкладки браузера</title>
  </head>
  <body>
    <ul class="site-navigation">
      <li class="popular"><p class="popular__item">Информация</p></li>
      <li><p class="title">Новости</p></li>
      <li id="banner"><h2 id="title">Новости</h2></li>
      <li><p class="popular__item"></p></li>
    </ul>
  </body>
</html>
```

popular__item img {...}

```
<!DOCTYPE html>
<html lang="ru">
  <head>
    <meta charset="utf-8">
    <title>Заголовок вкладки браузера</title>
  </head>
  <body>
    <ul class="site-navigation">
      <li class="popular"><p class="popular__item">Информация</p></li>
      <li><p class="title">Новости</p></li>
      <li id="banner"><h2 id="title">Новости</h2></li>
      <li><p class="popular__item"></p></li>
    </ul>
  </body>
</html>
```

.popular .popular__item {...}

```
<!DOCTYPE html>
<html lang="ru">
  <head>
    <meta charset="utf-8">
    <title>Заголовок вкладки браузера</title>
  </head>
  <body>
    <ul class="site-navigation">
      <li class="popular"><p class="popular__item">Информация</p></li>
      <li><p class="title">Новости</p></li>
      <li id="banner"><h2 id="title">Новости</h2></li>
      <li><p class="popular__item"></p></li>
    </ul>
  </body>
</html>
```



Псевдоклассы

- ❖ Усиливают обычные селекторы
- ❖ Позволяют выбирать элементы с учетом:
- ❖ Состояния
- ❖ Расположения относительно других элементов

`:hover` `:first-child`

`:focus` `:last-child`

`:active` `:nth-child()`

`:link` `:root`

`:visited`

`:valid`

`:invalid`

`:required`

`:optional`

`:checked`

`:disabled`

`:enabled`



Псевдоклассы

`/* Наведение на ссылку */`

`a: hover {...}`

`/* Каждый второй элемент списка */`

`li: nth-child(2n) {...}`

`/* Обязательные поля ввода */`

`input: required {...}`

`/* Отключенные кнопки */`

`button: disabled {...}`



Псевдоэлементы

- ❖ Создают виртуальные теги
- ❖ Позволяют стилизовать их

`::before`

`::after`

`::first-letter`

`::first-line`

Псевдоэлементы

```
<li class="header-login">
  <a href="#"></a>
</li>
<li class="header-login">
  <::before>Иконка перед
  ссылкой</::before>
  <a href="#"></a>
</li>
<li class="header-login">
  <a href="#"></a>
  <::after>Иконка после ссылки</::after>
</li>
```

```
.header-login::before {
  content: "Иконка перед
  ссылкой";
}
.header-login::after {
  content: "Иконка после
  ссылки";
}
```




CSS-свойства

position top left z-index

display width height margin padding

border

font-family font-size line-height color

text-decoration text-transform

background box-shadow text-shadow

/* Их очень и очень много */



Классификация свойств

- ❖ Оформление текста
- ❖ Позиционирование
- ❖ Блочная модель и поток
- ❖ Декоративные эффекты
- ❖ Анимация и динамические эффекты
- ❖ И другие



Каскадность

- ❖ К одному и тому же элементу в HTML может применяться несколько CSS-правил
- ❖ В этом случае свойства правил комбинируются

```
<li class="header-login">
```

```
  <a href="#"></a>
```

```
</li>
```

```
.header-login {  
  color: red;
```

```
}
```

```
p {  
  font-size: 18px;
```

```
}
```



Конфликт свойств

- ❖ Когда в CSS-правилах значения одного и того же свойства у одного элемента отличаются, то они конфликтуют

```
<li class="header-login header-block">  
  <a href="#"></a>  
</li>
```

```
.header-login {  
  color: red;  
}
```

```
.header-block {  
  color: blue;  
}
```



Специфичность

- ❖ Специфичность определяется по селектору
- ❖ Если она одинаковая, то побеждает то правило, которое расположено ниже в коде
- ❖ Чем проще селекторы, тем меньше проблем с расчётом специфичности

Специфичность

Вес селекторов (по убыванию):

<code>style=""</code>	1,0,0,0
<code>#id</code>	0,1,0,0
<code>.class</code>	0,0,1,0
<code>[attr=value]</code>	0,0,1,0
<code>LI</code>	0,0,0,1
<code>*</code>	0,0,0,0

У стилей, заданных в атрибуте `style`, на первой позиции будет единица — 1,0,0,0. Это самая высокая специфичность, которая перевешивает свойства, заданные другими способами.

Переопределить стили, заданные в `style`, можно дописав `!important` к значению свойства в таблице стилей.

Обратный вариант — универсальный селектор `*`. Он не имеет веса: 0,0,0,0.



Специфичность

`.popular-item {...} > p {...}`

`p.popular-item {...} > .popular-item {...}`

`p.popular-item {...} = .popular-item p {...}`

`.popular-item {...} < .popular-item .index {...}`

`#title {...} > .popular-item .index {...}`

`#title a {...} > #title {...}`



Типы значений

- ❖ Абсолютные
- ❖ Относительные
- ❖ Ключевые слова
- ❖ Цвета
- ❖ Функции
- ❖ Произвольные строки



Абсолютные значения

```
width: 1200px;
```

```
font-size: 16px;
```

```
padding-left: 15px;
```



Относительные значения

`width: 50%;` /* От ширины родителя */

`width: 100vw;` /* От ширины вьюпорта (окна) */

`width: 100vh;` /* От высоты вьюпорта (окна) */

`font-size: 2em;` /* От размера шрифта родителя */

`font-size: 2rem;` /* От размера шрифта body */



Ключевые слова

`text-transform: uppercase;`

`text-align: center;`

`display: flex;`

`color: red;`



Цвета

color: #f55;

color: #ff5555;

color: rgb(255, 85, 85);

color: rgba(255, 85, 85, 0.5);

color: hsl(0, 100%, 67%);

color: hsla(0, 100%, 67%, 0.5);



Функции

/* Получает содержимое атрибута */

content: attr(href);

/* Математические расчёты */

width: calc(100% - 250px);

/* Линейные градиенты */

background-image: linear-gradient(45deg, yellow, red);



Произвольные строки

```
font-family: "Arial", sans-serif;  
content: "Привет!";
```



CSS-директивы

```
@font-face {  
    font-family: "Arial";  
    src:  
        url("Arial.woff2") format("woff2"),  
        url("Arial.woff") format("woff");  
}
```

```
@media (max-width: 720px) {  
    .title {  
        display: none;  
    }  
}
```



Нормализация стилей

Normalize.css - это небольшой CSS-файл, который обеспечивает для HTML-элементов лучшую кроссбраузерность в стилях по умолчанию.

<http://necolas.github.io/normalize.css/>



Цели нормализации стилей

- Сохранить полезные настройки браузера, а не стирать их
- Нормализовать стили для широкого круга HTML-элементов
- Корректировать ошибки и основные несоответствия браузера



Темы раздела

- ❖ **Введение в CSS**
- ❖ Подключение шрифтов
- ❖ Сетки
- ❖ Flexbox
- ❖ БЭМ



Подключение шрифтов



Алгоритм работы со шрифтов

1. Проверяем, стандартный ли шрифт
<https://www.cssfontstack.com/>
2. Если нет, ищем на внешних сервисах
3. Скачиваем, конвертируем в форматы woff и woff2
4. Подключаем локально



Заключение по шрифтам

- ❖ Основные форматы - woff и woff2
- ❖ Чем меньше нестандартных шрифтов и их вариаций, тем лучше
- ❖ Иногда лучше вставить надпись картинкой, чем подключать шрифт
- ❖ Каждая градация жирности - отдельный подключаемый шрифт
- ❖ Не полагайтесь на возможности браузера (жирность и наклонное начертание)



Темы раздела

- ❖ **Введение в CSS**
- ❖ **Подключение шрифтов**
- ❖ Сетки
- ❖ Flexbox
- ❖ БЭМ



Базовые критерии

- ❖ Вся собственная стилизация выполнена в одном файле
- ❖ В CSS отсутствует `!important`
- ❖ Для стилизации не использованы `id`
- ❖ Подключены правильные шрифты, их размеры, высота строк, цвет и толщина как на макете
- ❖ Указаны альтернативные варианты шрифта и тип семейства в `font-family`
- ❖ Единообразное написание и форматирование кода в HTML, CSS и JS



Дополнительные критерии

- ❖ Подключен `normalize.css`
- ❖ Нет вложенности селекторов больше двух
- ❖ Нет глобальных стилей тегов



Сетки

СТУДИЯ ДИЗАЙНА

Дизайн & Полиграфия & Наружная реклама

Расчет стоимости

ИМЯ

ТЕЛЕФОН

ПОЗВОНИТЕ НАМ

✓ Согласен с правилами обработки данных, принимаю условия конфиденциальности

Мы все это делаем!

ДИЗАЙН | ПОЛИГРАФИЯ | НАРУЖНАЯ РЕКЛАМА | 3D РЕКЛАМА | ПРЕСС-ФОТО



ЛОГОТИП

Логотип представляет собой название, графический элемент или комбинацию из них, способствующую быстрому запоминанию.



ФИРМЕННЫЙ СТИЛЬ

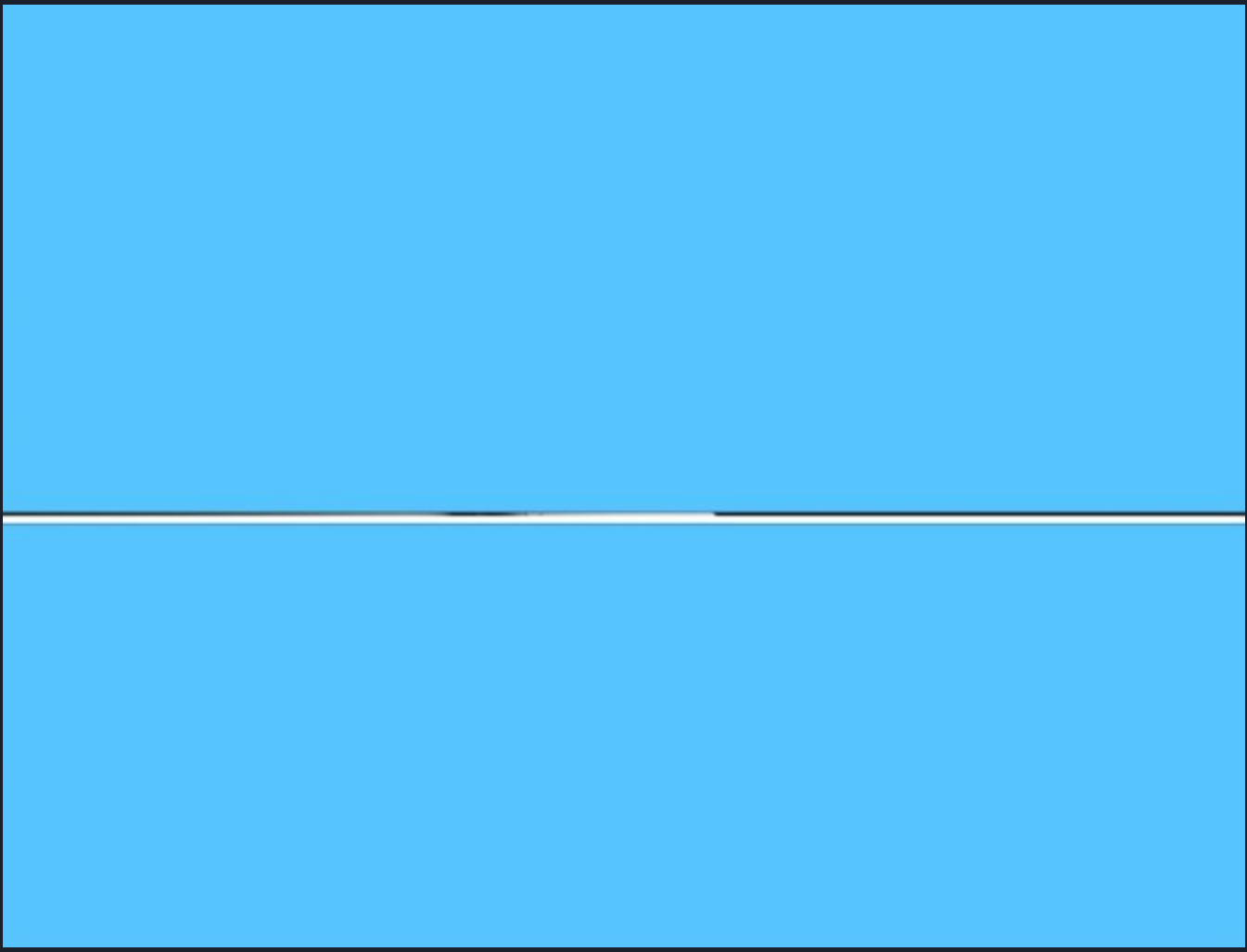
Выражается в атрибутах корпоративного стиля. То же, что и идентичность бренда, только по отношению к корпоративному бренду.

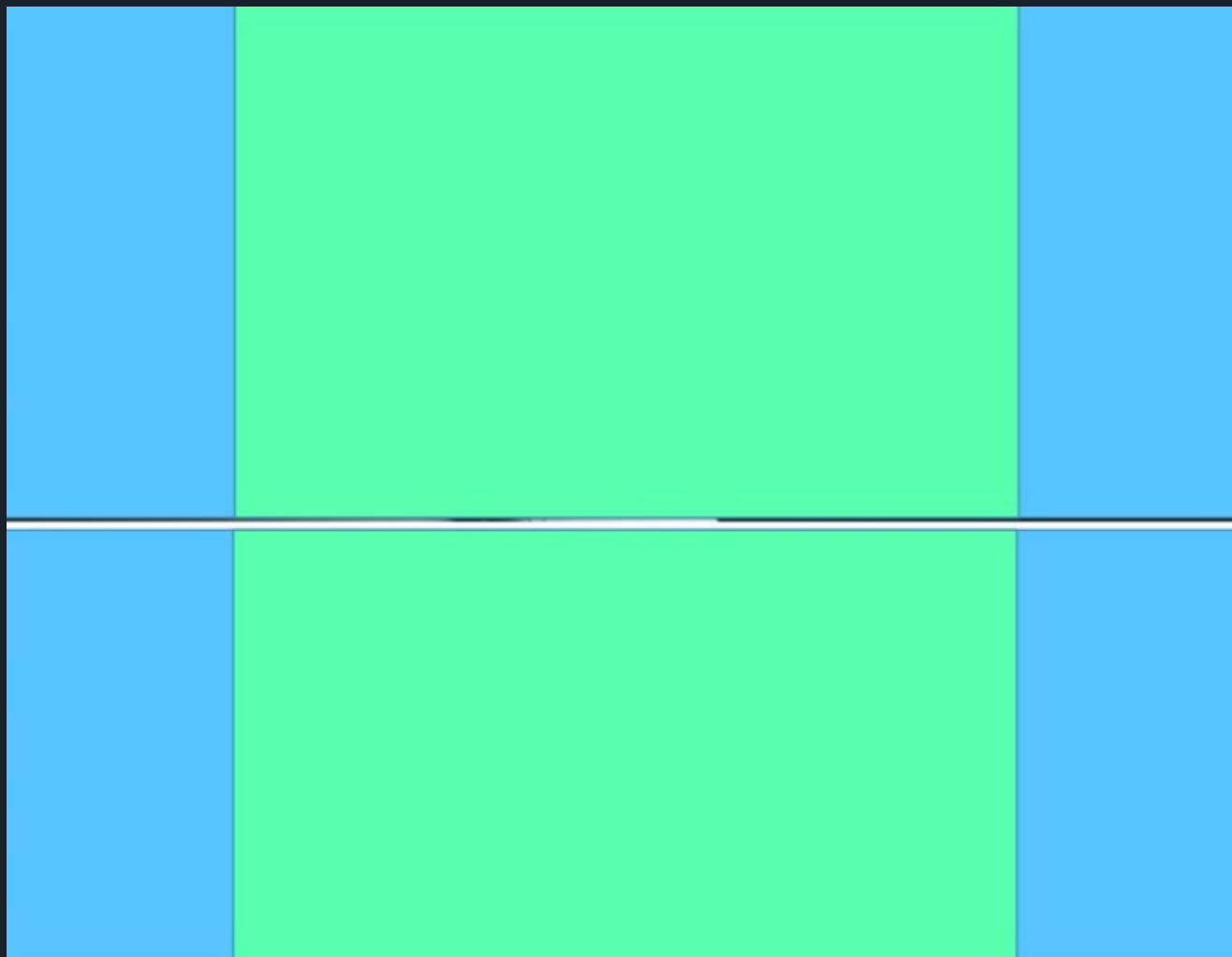


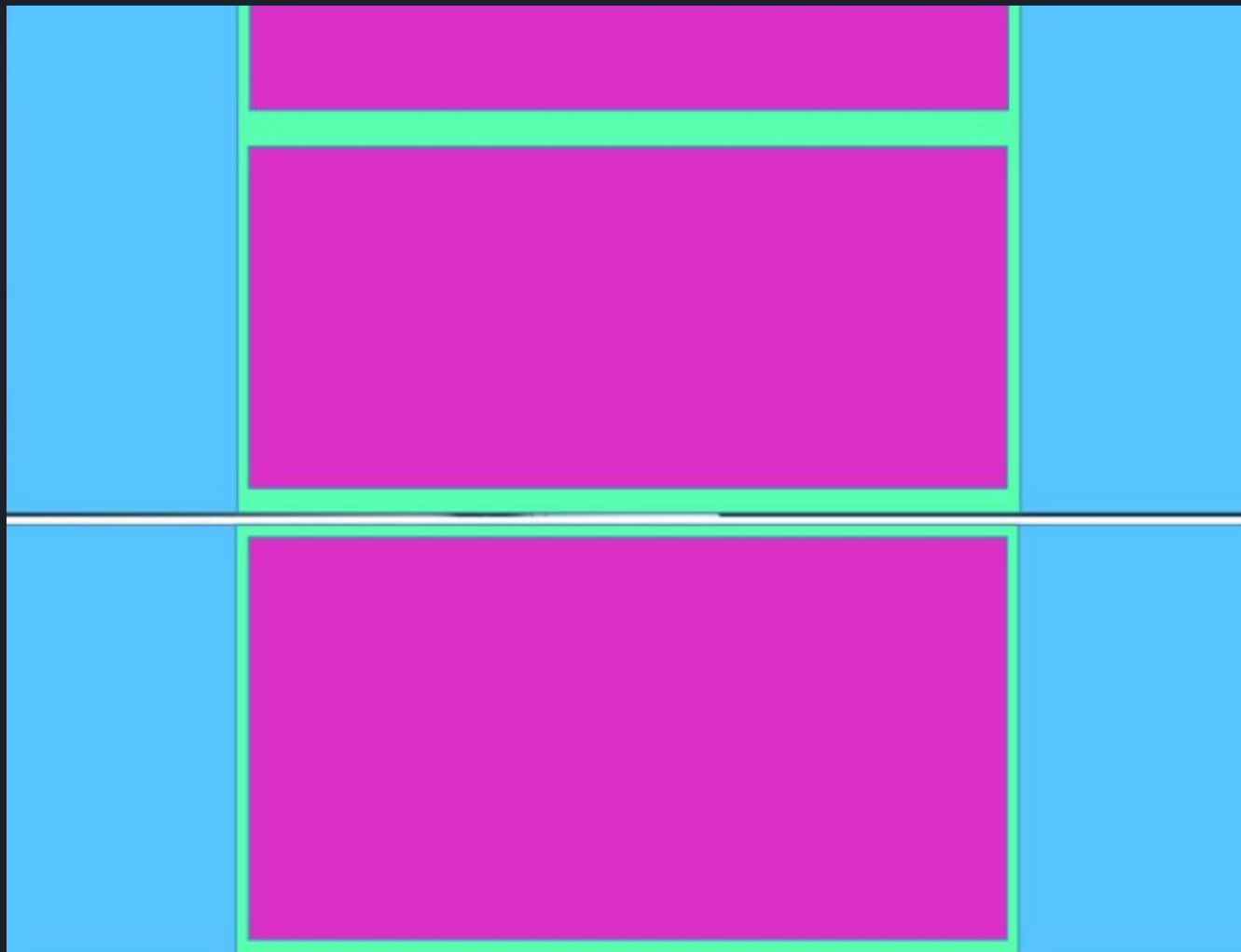
ДИЗАЙН САЙТА

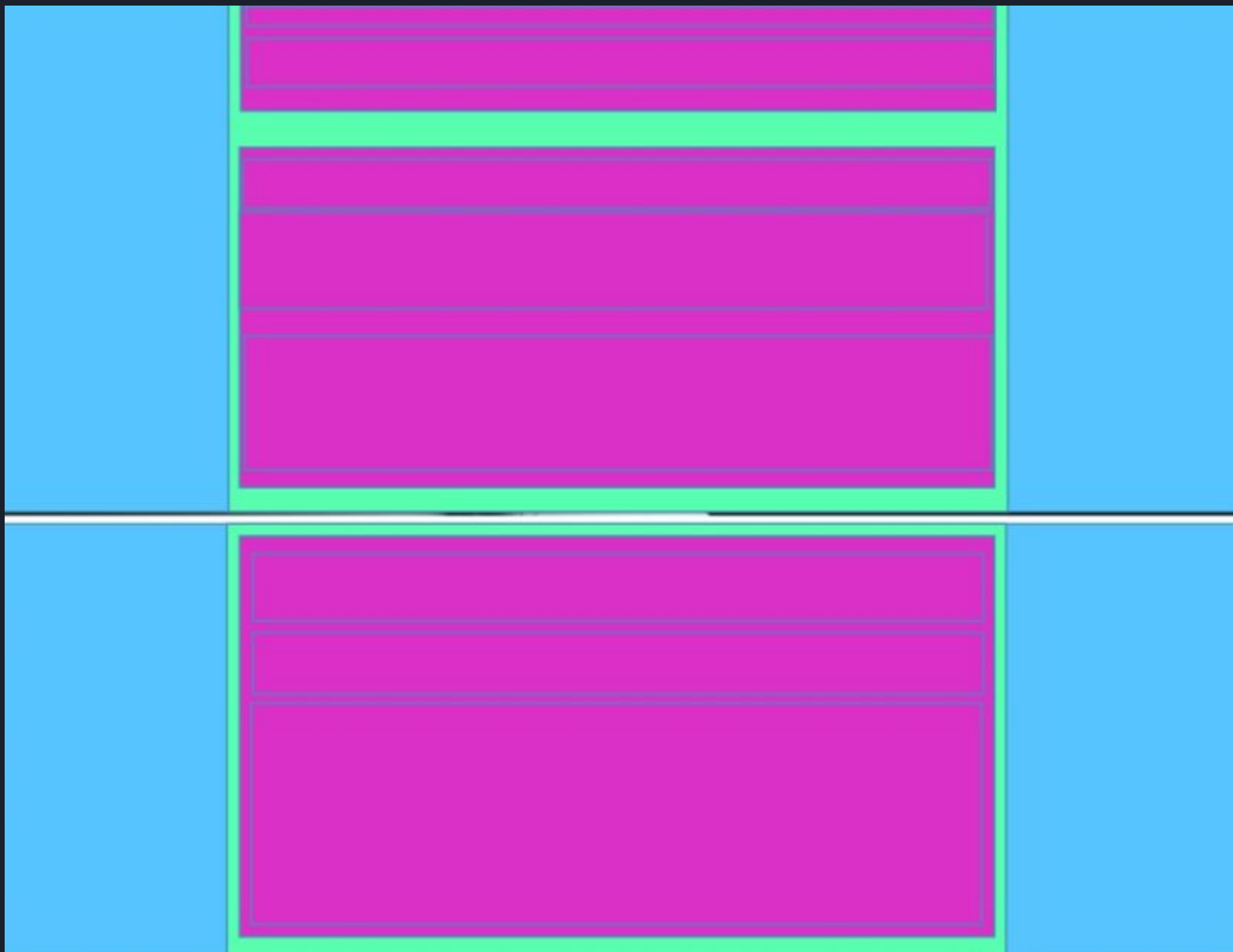
Результатом дизайна, в ходе которого создается прототип, является веб-интерфейс для сайта.

ЗАКАЗАТЬ УСЛУГУ











Что такое поток документа?

<body>

<header class="header"><header>

<section class="column1"><section>

<section class="column2"><section>

<section class="column3"><section>

<footer class="footer"><footer>

</body>

.header

.column1

.column2

.column3

.footer



Что такое поток документа?

```
<body>
```

```
<header class="header"><header>
```

```
<div class="column1"><div>
```

```
<div class="column2"><div>
```

```
<div class="column3"><div>
```

```
<footer class="footer"><footer>
```

```
</body>
```

.header

.column1

.column2

.column3

.footer



Что такое поток документа?

<body>

<header class="header"><head

<div class="column1"><div>

<div class="column2"><div>

<div class="column3"><div>

<footer class="footer"><footer>

<body>

.header

column1

column2

column3

.footer



Блочная модель документа

CSS-свойства, относящиеся к блочной модели:

- ❖ Изменяют размер элементов и занимаемое ими место
- ❖ Влияют на поведение элементов в потоке



Типы боксов

Тип бокса задается свойством `display`

- ❖ Блочные (`display: block`)
- ❖ Строчные (`display: inline`)
- ❖ Блочнo-строчные (`display: inline-block`)
- ❖ Флексы (`display: flex`)
- ❖ Гриды (`display: grid`)
- ❖ Табличные (`display: table`)
- ❖ Другие



Блочные боксы

Блочные боксы - прямоугольные области на странице

Блочными боксами по умолчанию обладают:
div, section, h1, h2, h3, h4, h5, h6, p, ul, ol и другие



Особенности блочных

- ❖ Принудительный перенос строки до и после
- ❖ Воспринимают ширину, высоту, внутренние и внешние отступы
- ❖ Занимают все доступное по ширине пространство
- ❖ По высоте подстраиваются под содержимое

Особенности блочных

itacts.scss × pages\offers-detail.html × sections\offers-detail.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
  <h1>Типы элементов</h1>
  <section>
    <h2>Блочные</h2>
    <p>Элементы</p>
  </section>
</body>
</html>
```

Типы элементов

Блочные

Элементы



Строчные боксы

Строчные боксы - фрагменты текста

Строчными боксами по умолчанию обладают:

`span`, `a`, `b`, `em`, `i`, `input` и другие



Особенности строчных

- ❖ Нет переносов строки до и после - располагаются в одной строке
- ❖ Ширина и высота зависят только от содержания, задать размеры с помощью CSS нельзя
- ❖ Воспринимают только горизонтальные отступы
- ❖ Ведут себя как текст

Особенности строчных

```
<!DOCTYPE html>
<html lang="en">
</head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
  <h1>Строчные элементы</h1>
  <p>
    <em>Строчные элементы</em> не создают перенос <span>строки</span> до и после себя. Такие элементы располагаются
    в строке слева направо.
    Если <span>строчный элемент не помещается в родительский контейнер</span>, то он переносится на следующую
    строку
  </p>
</body>
</html>
```

Строчные элементы

Строчные элементы не создают переносов строки до и после себя. Такие элементы располагаются в строке слева направо. Если строчный элемент не помещается в родительский контейнер, то он



Свойства блочной модели

Влияют на размер элемента:

- ❖ width
- ❖ height
- ❖ margin
- ❖ padding
- ❖ border



width, min-width, max-width

Ширина содержимого

```
width: 500px;
```

```
width: 50%;
```

```
width: auto; /* по умолчанию */
```



height, min-height, max-height

Высота содержимого

```
height : 500px;
```

```
height: 50%; /* использовать аккуратно */
```

```
height: auto; /* по умолчанию */
```



Свойства блочной модели

Блочные элементы по умолчанию занимают всю доступную ширину. Их высота зависит от содержания. Если задать фиксированную высоту и содержимое не будет в неё помещаться, оно просто «выпадет» из контейнера и перестанет «чувствоваться» другими элементами.

Размеры строчных элементов полностью зависят от их содержания и с помощью CSS не изменяются.



Советы

Старайтесь не использовать одновременно `width` и `height`, если это не декоративный элемент с фиксированными размерами.

Старайтесь не задавать фиксированную высоту. Если всё таки нужна фиксированная высота, то лучше использовать `min-height`.



padding

Внутренние отступы

```
padding: 10px;
```

```
padding: 5%;
```

```
padding: 2em;
```



padding с разных сторон

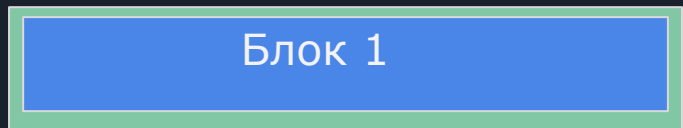
```
padding: 10px; /* одинаково со всех сторон */  
padding: 10px 20px; /* сверху+снизу, справа+слева */  
padding: 10px 20px 30px; /* сверху, справа+слева, снизу */  
padding: 10px 20px 30px 40px; /* сверху, справа, снизу, слева  
*/
```

```
padding-left: 10px;  
padding-right: 20px;  
  
padding-top: 30px;  
  
padding-bottom: 40px;
```

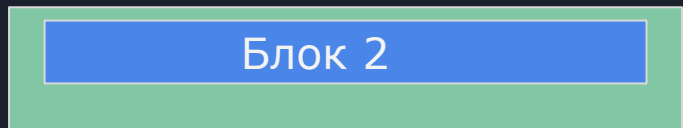



Примеры padding

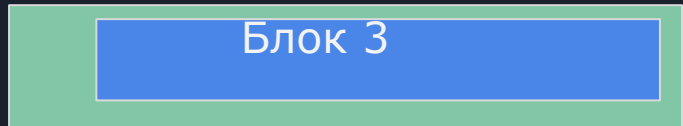
`padding: 20px;`



`padding: 20px;`
`padding-bottom: 40px;`



`padding: 10px 20px 30px 40px`





margin

Внешние отступы

```
margin: 10px;
```

```
margin: 5%;
```

```
margin: auto; /* магическое свойство, о нём позже */
```

```
margin: 2em;
```



margin с разных сторон

```
margin: 10px; /* одинаково со всех сторон */
```

```
margin: 10px 20px; /* сверху+снизу, справа+слева */
```

```
margin: 10px 20px 30px; /* сверху, справа+слева, снизу */
```

```
margin: 10px 20px 30px 40px; /* сверху, справа, снизу, слева */
```

```
margin-left: 10px;
```

```
margin-right: 20px;
```

```
margin-top: 30px;
```

```
margin-bottom: 40px;
```

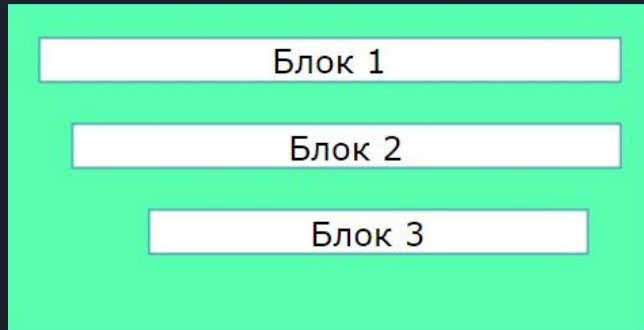
Примеры margin

`margin: 20px;`

`margin: 20px;`

`margin-left: 40px;`

`margin: 20px 40px 60px 80px;`





border

Рамка

Состоит из трех компонентов:

1. Ширина рамки - `border-width`
2. Стил ь рамки - `border-style`
3. Цвет рамки - `border-color`

border

Примеры:

`border: 6px solid red;`


`border-top: 1px solid red;`

`border-left: 2px dashed blue;`

`border-bottom: 3px dotted green;`

`border-right: 4px groove yellow;`





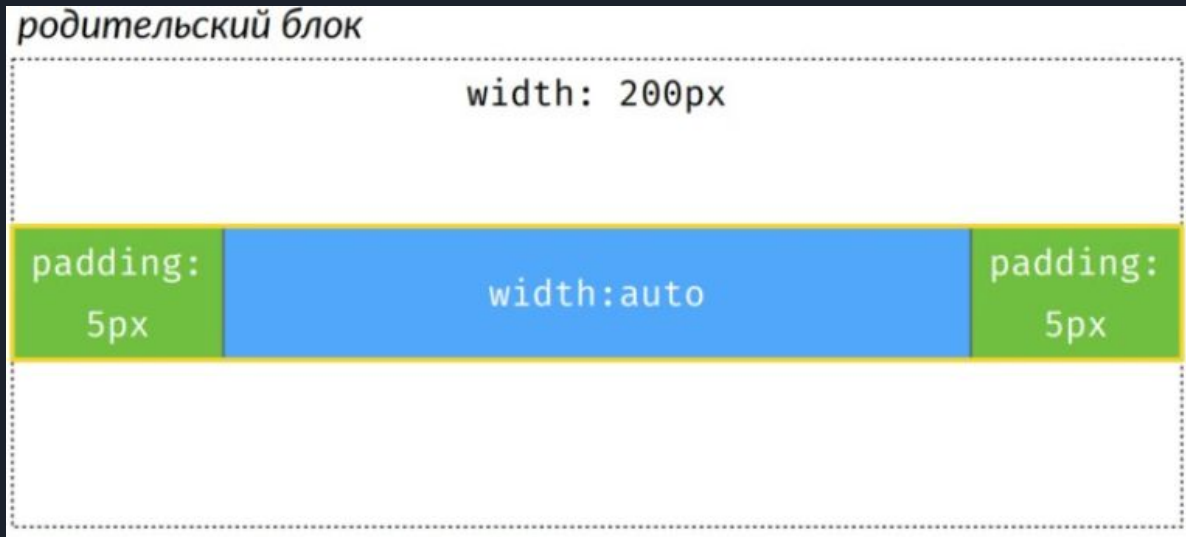
Расчет полного размера блочного бокса

Если ширина не задана, общая ширина равна доступному месту в родителе.



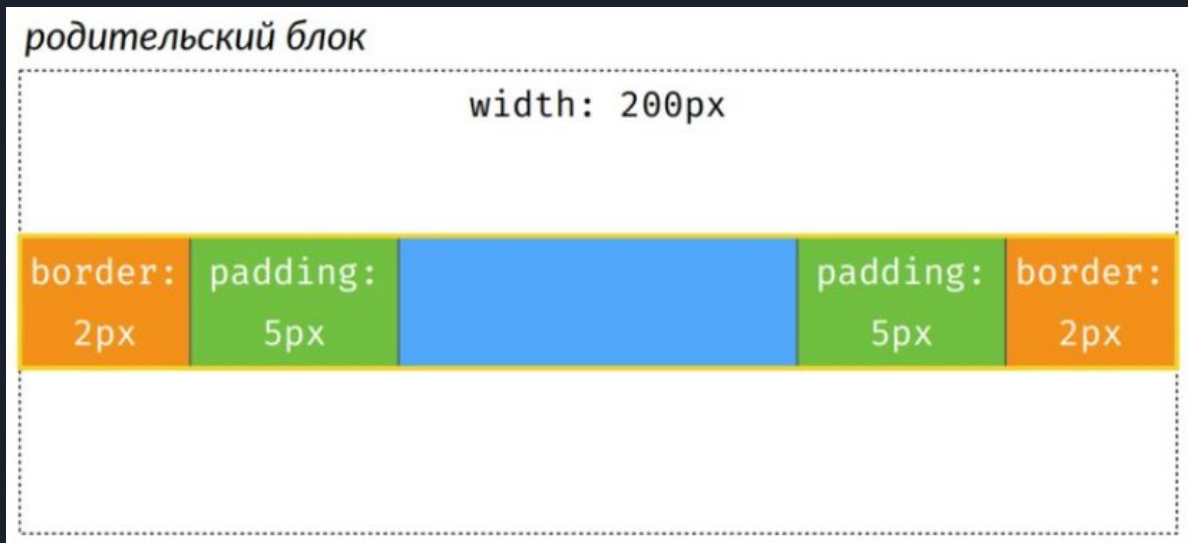
Расчет полного размера блочного бокса

При добавлении внутренних отступов ужимается содержимое.



Расчет полного размера блочного бокса

Рамки также приводят к уменьшению области содержимого



Расчёт полного размера блочного бокса

Внешние отступы изменяют общую ширину блока.



Расчёт полного размера блочного бокса


Если ширина задана, то общая ширина равна: ширина контента + внутренние отступы + ширина рамок



Расчёт полного размера блочного бокса


Внешние отступы не влияют на общую ширину блока с заданной шириной.





Расчёт полного размера блочного бокса

1. Если `height: auto`, то общая высота равна:
высота содержимого + внутренний отступ + рамки
2. Если высота задана конкретно, то общая высота равна:
заданная высота + внутренний отступ + рамки



Расчёт полного размера блочного бокса

Общая ширина и высота бокса - это не то же самое,
что свойства `width` и `height`

Это поведение можно изменить

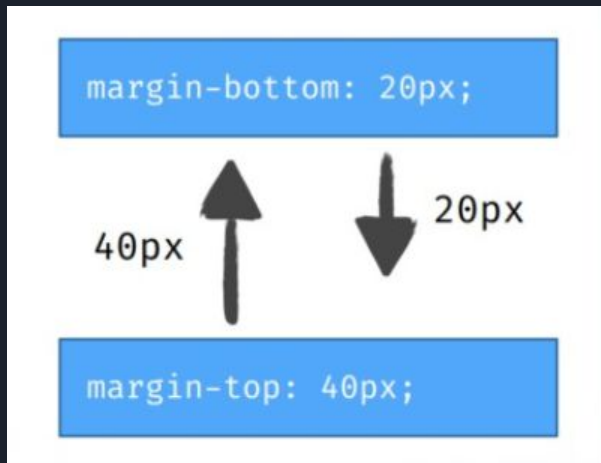


Тонкости блочной модели

- ❖ Схлопывание внешних отступов (`margin`)
- ❖ Как расположить элемент по центру
- ❖ Ширина по умолчанию и 100%
- ❖ `box-sizing`

Схлопывание внешних отступов

В вертикальном направлении внешние отступы (`margin`) не складываются, а выбирается максимальный из доступных.



Как расположить элемент по центру?

`margin-left: auto;`

`margin-right: auto;`

`margin: 0 auto;`



width: auto и width: 100%

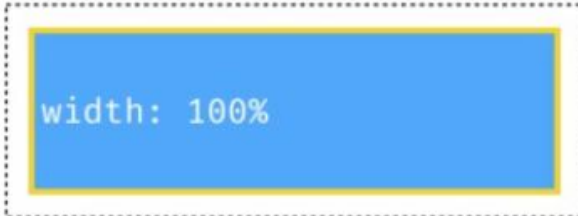
родительский блок

`width: auto`

A diagram illustrating the 'width: auto' property. It shows a blue rectangular box with a yellow border. The text 'width: auto' is written inside the box. This box is contained within a larger, dashed-line rectangular box representing the parent container.

родительский блок

`width: 100%`

A diagram illustrating the 'width: 100%' property. It shows a blue rectangular box with a yellow border that fills the entire width of the parent container. The text 'width: 100%' is written inside the box. The parent container is represented by a dashed-line rectangular box.

родительский блок

`width: auto`

`padding`

A diagram illustrating the 'width: auto' property in the presence of padding. It shows a blue rectangular box with a yellow border and a green rectangular box with a black border. The blue box is labeled 'width: auto' and the green box is labeled 'padding'. Both are contained within a dashed-line rectangular box representing the parent container. The blue box is positioned to the left of the green box, and they are separated by a vertical line.

родительский блок

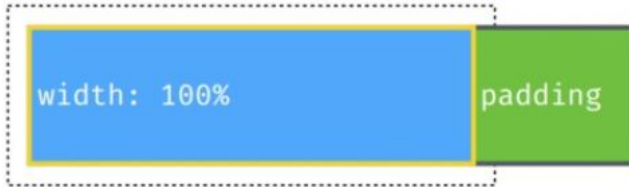
`width: 100%`

`padding`

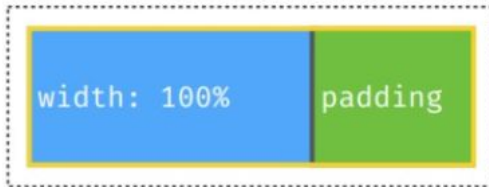
A diagram illustrating the 'width: 100%' property in the presence of padding. It shows a blue rectangular box with a yellow border and a green rectangular box with a black border. The blue box is labeled 'width: 100%' and the green box is labeled 'padding'. Both are contained within a dashed-line rectangular box representing the parent container. The blue box is positioned to the left of the green box, and they are separated by a vertical line. The blue box fills the width of the parent container minus the padding.

box-sizing

`box-sizing: content-box;`



`box-sizing: border-box;`





Content

The content of the box, where text and images appear.



Padding

Clears an area around the content.



Border

A border that goes around the padding.



Margin

Clears an area around the border.



Without box-sizing: border-box

Margin, borders and padding are drawn outside the set width of your content.



With box-sizing: border-box

Borders and padding are drawn inside the set width of your content. The margin is drawn outside.



Темы раздела

- ❖ **Введение в CSS**
- ❖ **Подключение шрифтов**
- ❖ **Сетки**
- ❖ Flexbox
- ❖ БЭМ



Flexbox

Исходное состояние

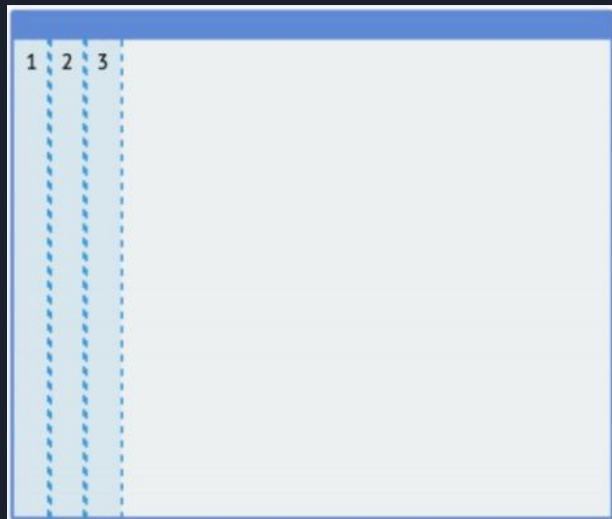
```
<section class="flex-container">
  <div class="flex-container__item">1</div>
  <div class="flex-container__item">2</div>
  <div class="flex-container__item">3</div>
</section>
```



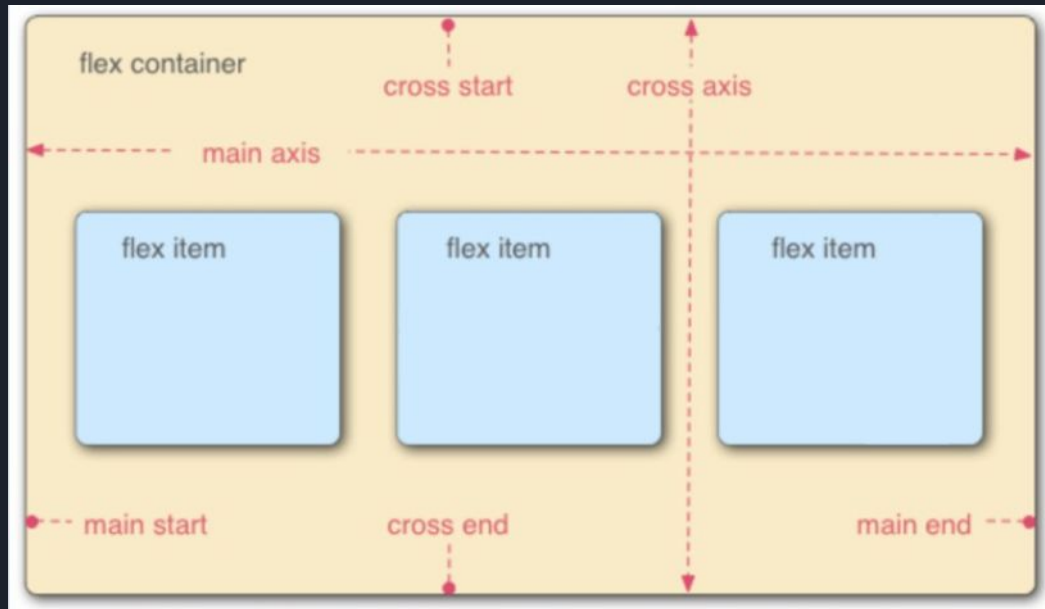
Включаем флекс

```
<section class="flex-container">
  <div class="flex-container__item">1</div>
  <div class="flex-container__item">2</div>
  <div class="flex-container__item">3</div>
</section>
```

```
.flex-container {
  display: flex;
}
```

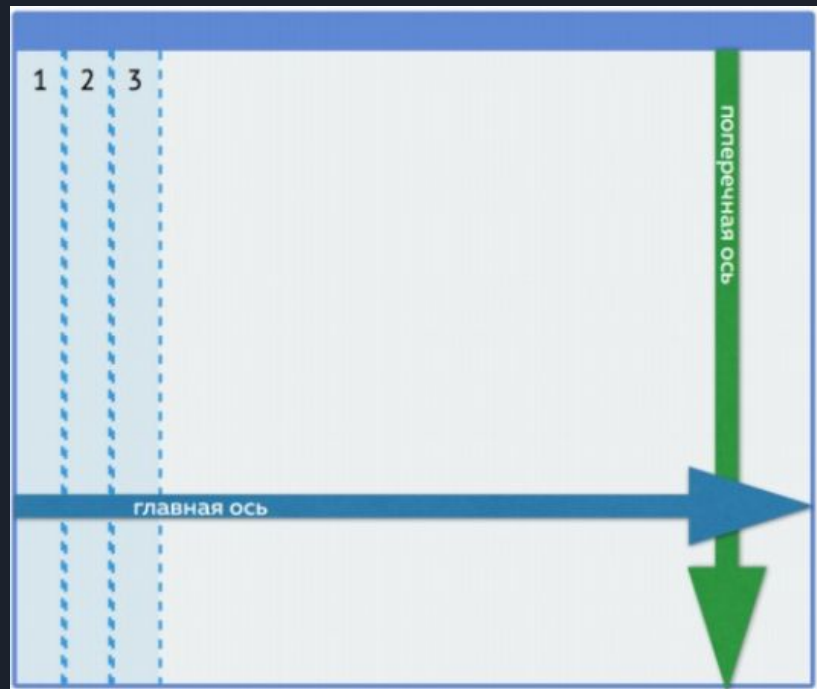


Терминология

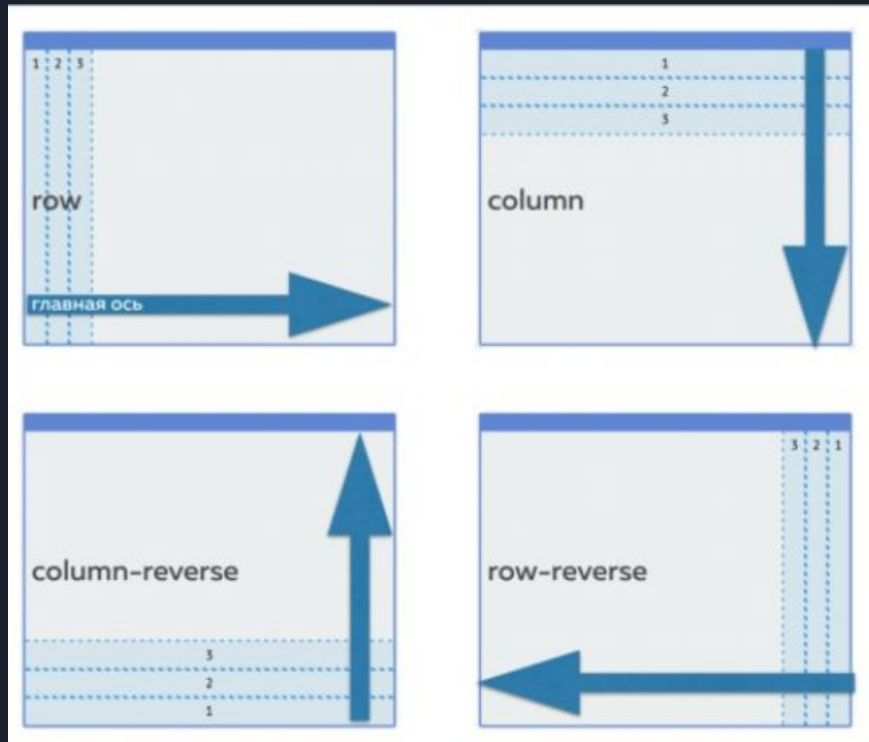


Направление главной оси

```
.flex-container {  
  display: flex;  
  flex-direction: row; /* default */  
  //flex-direction: row-reverse;  
  //flex-direction: column;  
  //flex-direction: column-reverse;  
}
```



flex-direction





Флекс и блочная модель

- ❖ Размер флекс-элементов рассчитывается как в обычной блочной модели: размер содержимого + отступы + рамки
- ❖ Работает box-sizing
- ❖ Флекс-элементы по умолчанию сжимаются под содержимое
- ❖ Внешние отступы не схлопываются



margin: auto

- ❖ Работает во всех направлениях
- ❖ «Съедает» свободное место
- ❖ Если у нескольких элементов есть автоматические отступы, то свободное место делится поровну

margin: auto

```
.flex-container {  
  display: flex;  
  flex-direction: row; /* default */  
  //flex-direction: row-reverse;  
  //flex-direction: column;  
  //flex-direction: column-reverse;  
}  
.flex-container__item:nth-child(2) {  
  margin-top: auto;  
  margin-left: auto;  
}
```





Выравнивание элементов

- ❖ Вдоль главной оси
- ❖ Вдоль поперечной оси



Выравнивание вдоль главной оси

2 КАРТИНКИ

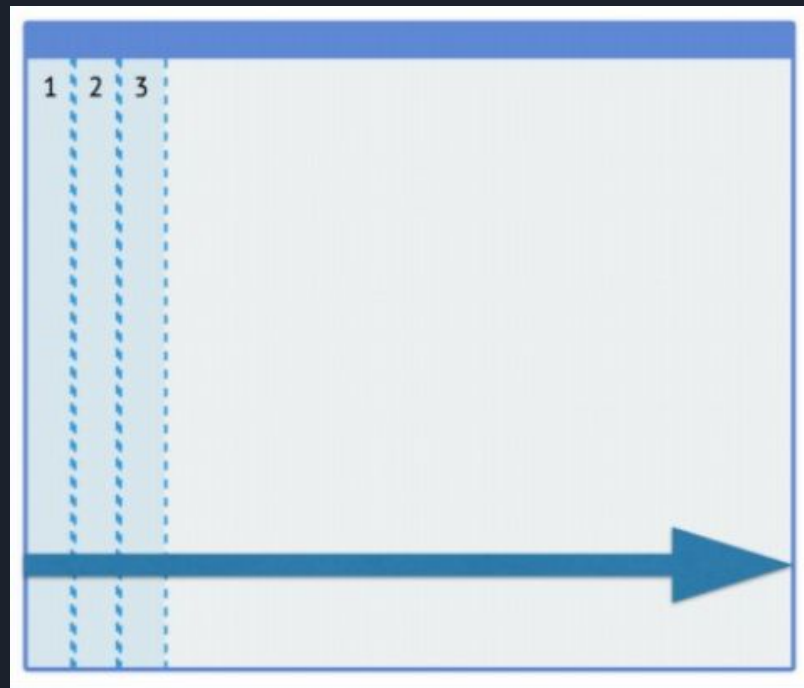


justify-content

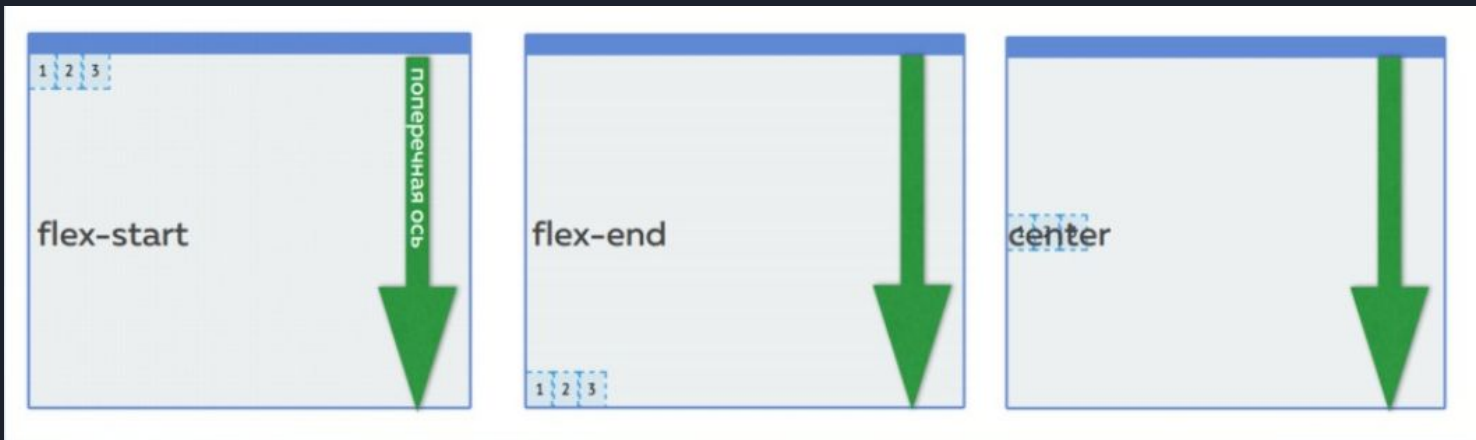
КАРТИНКА

Выравнивание вдоль поперечной оси

```
.flex-container {  
  display: flex;  
  flex-direction: row;  
  justify-content: flex-start; /* По умолчанию */  
  //justify-content: flex-end;  
  //justify-content: center;  
  //justify-content: space-between;  
  //justify-content: space-around;  
}
```

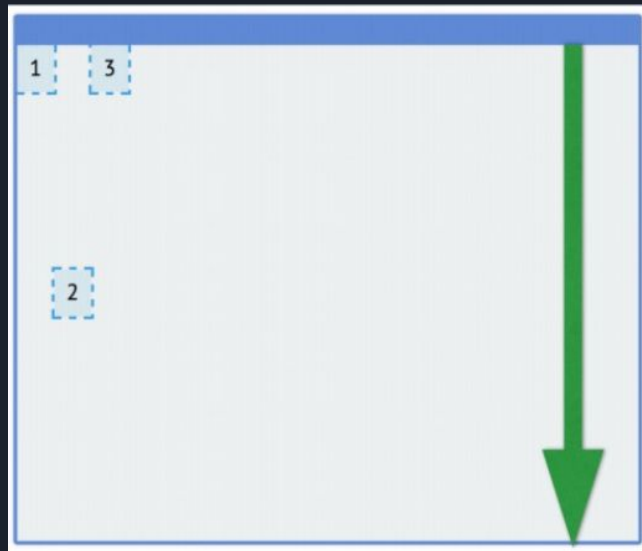


align-items



align-self

```
.flex-container {  
  display: flex;  
  flex-direction: row;  
  align-items: flex-start;  
}  
.flex-container__item:nth-child(2) {  
  align-self: center;  
}
```





Построение сеток на флексблоках

- ❖ Всегда задавайте ширину блоков
- ❖ Расстояние между колонками можно задавать:
- ❖ С помощью `justify-content`, если отступы одинаковые
- ❖ С помощью `margin`, если отступы разные
- ❖ Следите за псевдоэлементами у флекс-контейнера



Базовые критерии

- ❖ Раскладка блоков на странице сделана с помощью `flex-layout`.



Дополнительные критерии

- ❖ Верстка проходит тест на переполнение контентом



Темы раздела

- ❖ **Введение в CSS**
- ❖ **Подключение шрифтов**
- ❖ **Сетки**
- ❖ **Flexbox**
- ❖ БЭМ



БЭМ



БЭМ

- ❖ Набор абстракций для разработки интерфейса
- ❖ Появился от необходимости независимых блоков
- ❖ Есть большой БЭМ (стэк) и малый БЭМ (стили)
- ❖ Самая популярная методология



Коротко о БЭМ

- ❖ Только классы
- ❖ Без вложенности



Нотация

```
.block
.block_modifier
.block__element
.block__element_modifier

.block
.block--modifier
.block__element
.block__element--modifier

.block
.block._modifier
.block__element
.block__element._modifier
```



Tab 1 Tab 2 **Tab 3** Tab 4

Logo

Search

Login

Password

Sign in

Tab 1 Tab 2 **Tab 3** Tab 4

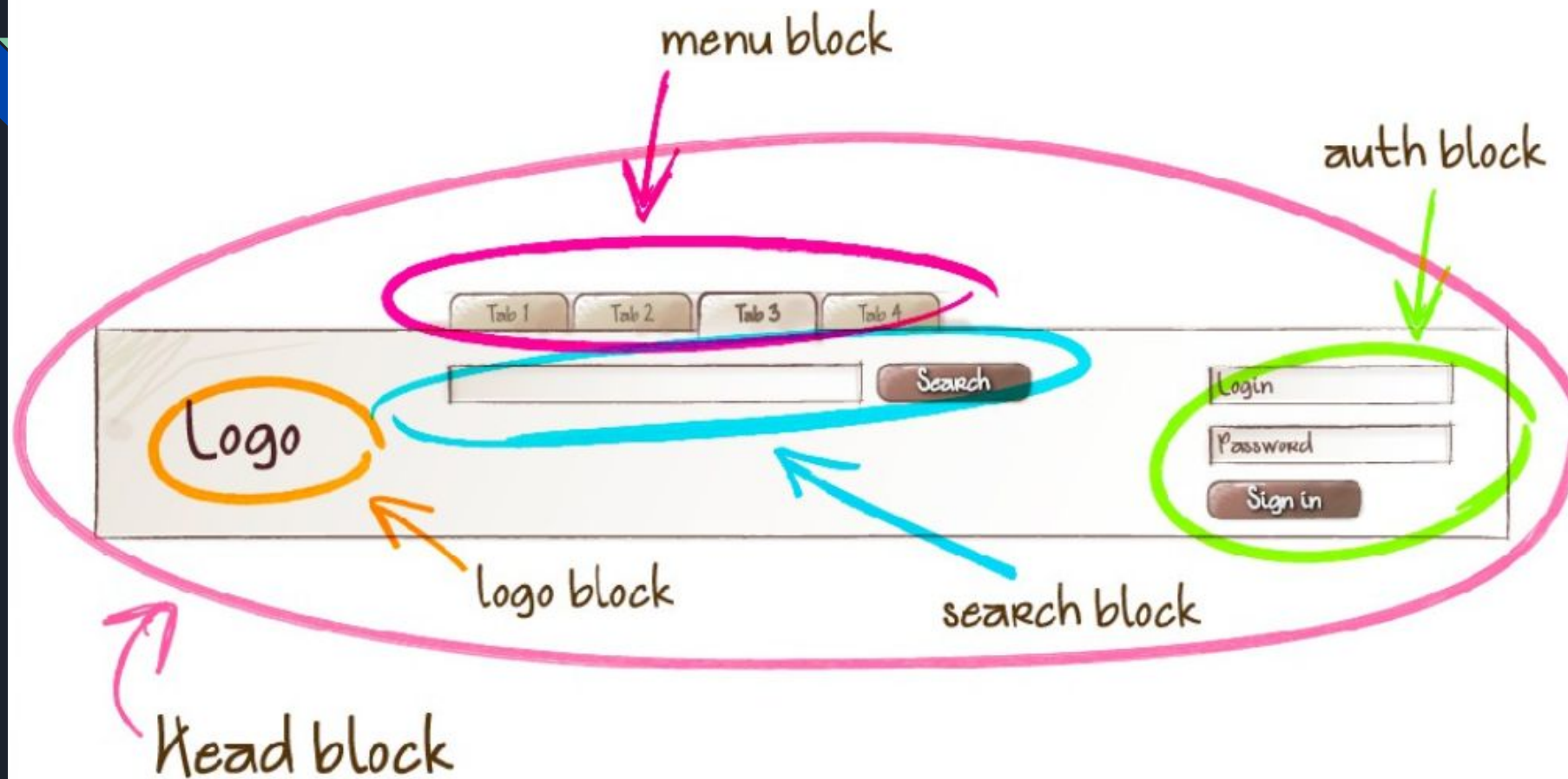
Login

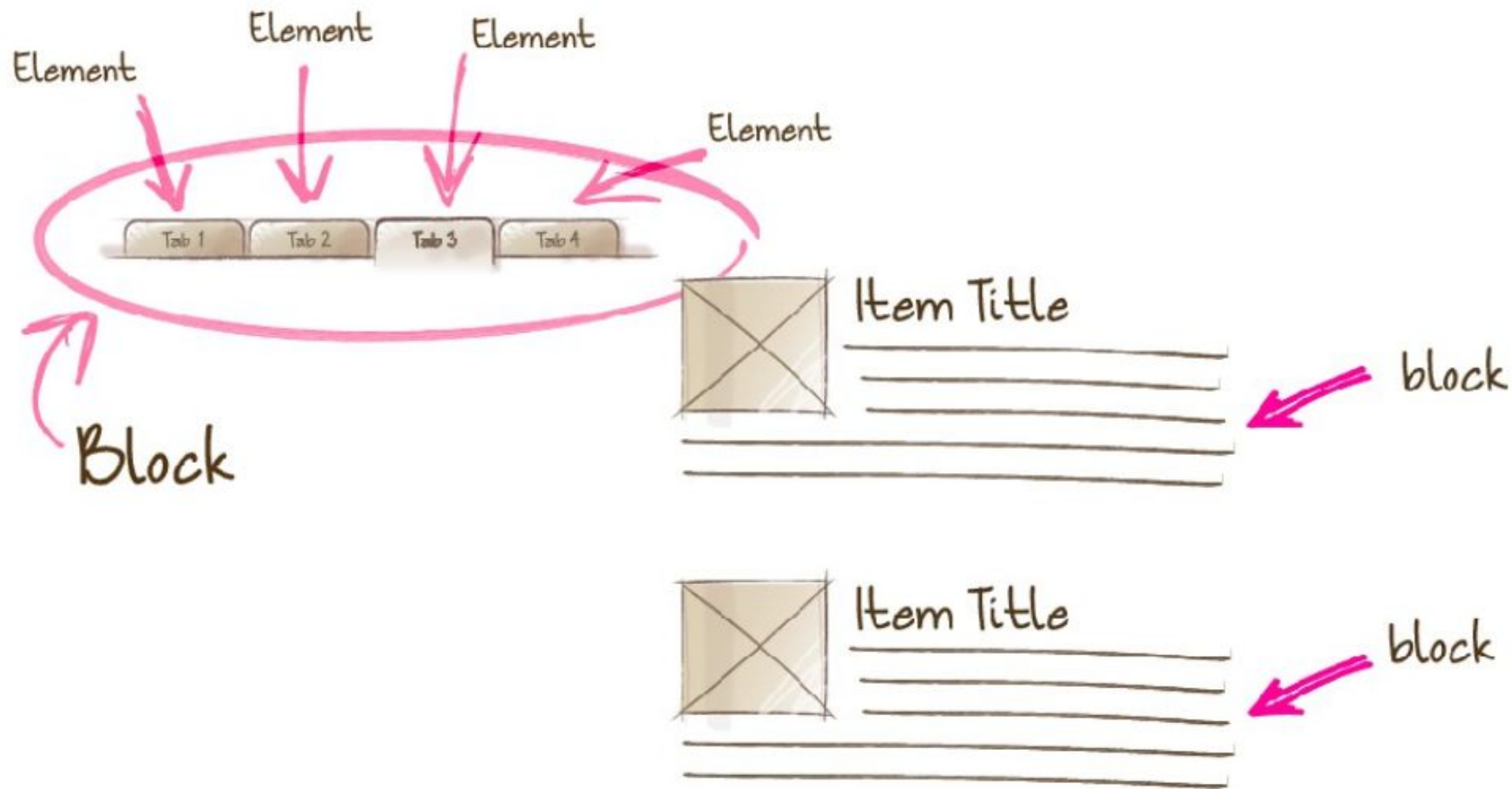
Password

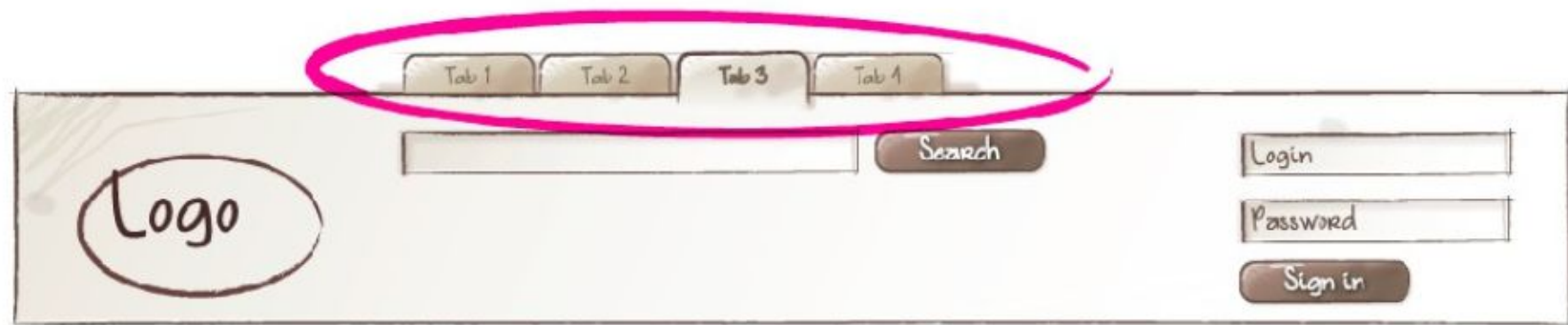
Sign in

Search

Logo







Page Title

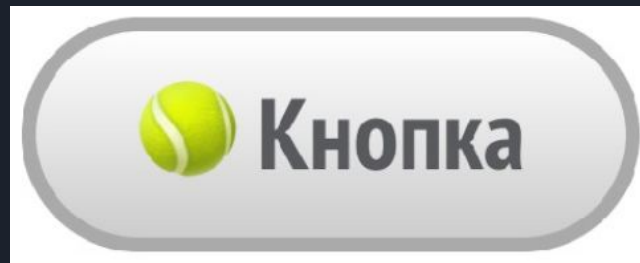
<hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>	<hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>	<hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>
---	---	---



Кнопка

```
<button type="button" class="button">  
  <span class="button__icon"></span>  
  <span class="button__text">Кнопка</span>  
</button>
```

```
.button {  
  background-color: gray;  
}  
.button__icon {  
  background-image: url('icon.png');
```



Кнопка

```
<button type="button" class="button button_red">  
  <span class="button__icon"></span>  
  <span class="button__text">Кнопка</span>  
</button>
```

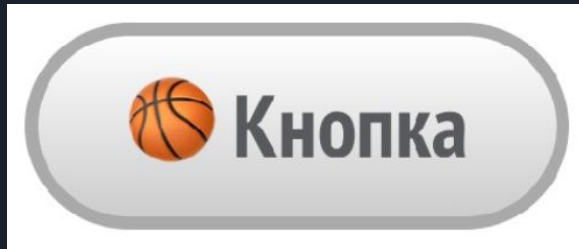
```
.button {  
  background-color: gray;  
}  
.button_red {  
  background: red;  
}  
.button__icon {  
  background-image: url('icon.png');  
}
```



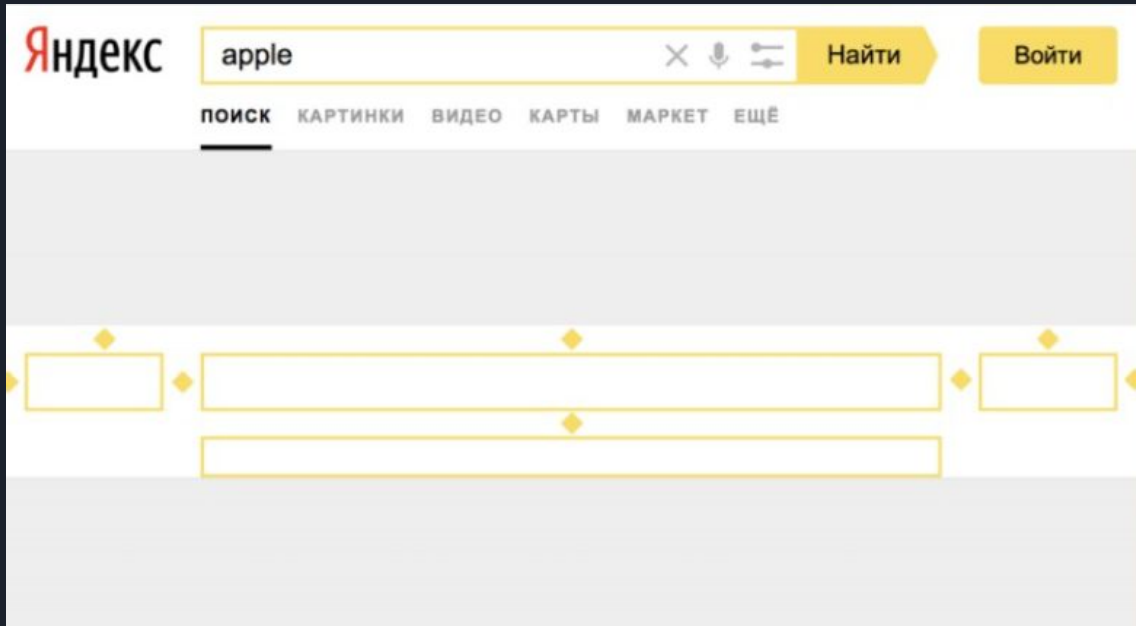
Кнопка

```
<button type="button" class="button">
  <span class="button__icon button__icon_basket"></span>
  <span class="button__text">Кнопка</span>
</button>
```

```
.button {
  background-color: gray;
}
.button_red {
  background: red;
}
.button__icon {
  background-image: url('icon.png');
}
.button__icon_basket {
  background-image: url('basketball-icon.png');
}
```



Миксы





Темы раздела

- ❖ **Введение в CSS**
- ❖ **Подключение шрифтов**
- ❖ **Сетки**
- ❖ **Flexbox**
- ❖ **БЭМ**



Базовые критерии

- ❖ Разметка должна быть сделана по БЭМ



Задание на дом

1. Обязательно посмотреть в справочнике, какие свойства существуют в CSS
2. Завершить разметку, добавив в неё сетки
 - ❖ Если получится, сделать базовую стилизацию сеток на флексах
 - ❖ Если не получится, сделаем на следующем занятии вместе
3. Полностью переписать все классы по БЭМ, если это не было выполнено
4. Подключить необходимые шрифты
5. Сделать стилизацию текстовых блоков (шрифт, размер шрифта, высоту строки, цвет текста)
6. Почитать и перечитать статьи, которые будут выложены и были выложены ранее