Бауманский учебный центр
Специалист.ru
Мировые стандарты обучения

Модуль 6. Iframe
Модуль 7. Формы для сбора данных
Модуль 8. Макетирование страниц

Басов Денис Алексеевич

specialist.ru

Описание курса

HTML и CSS. Уровень 1. Создание сайтов на HTML 5 и CSS 3

Тема	Часы
1. Введение в HTML	4
2. Структура страницы	2
3. Создание гиперссылок	2
4. Основы CSS	4
5. Размещение изображений, списков и таблиц	4
6. Iframe	1
7. Формы для сбора данных	3
8. Макетирование страницы с CSS	4
9. Таблицы стилей для печати и мобильной версии сайта	4
10. Подготовка, размещение и поддержка сайта	3
11. Дополнительная информация	1

Программа курса предусматривает лабораторные работы по каждой теме, а также выполнение домашних заданий и контроль знаний

В этом модуле мы рассмотрим:

- Взаимодействие ссылок с iframe-элементами
- Встраивание внешних файлов с помощью элемента <iframe>
- Цель и состав форм на веб-страницах
- Различные элементы формы и их атрибуты
- Создание формы обратной связи и обработка данных
- Изменение оформления границ и отступов элементов.
- Задание ширины и высоты элементов.
- Использование свойства float для плавающих элементов.
- Управление видимостью элементов.
- Применение различных значений свойства display для сложных макетов.

<iframe> - встраивание внешнего контента

Элемент iframe в HTML позволяет встраивать на веб-страницу документы, видео, интерактивные медиафайлы и другие части содержимого из других источников. Это один из старейших HTML-элементов, впервые его поддержка была включена в браузер Microsoft Internet Explorer в 1997 году

Однако, стоит отметить, что использование iframe может привести к проблемам с безопасностью и производительностью. В некоторых случаях встраиваемый документ может оказывать определенное влияние на вашу страницу, например вызовет появление всплывающих окон, уведомлений или будет автоматически проигрывать видео

<iframe> - встраивание внешнего контента

```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Макетирование страниц</title>
    <link rel="stylesheet" href="style.css" />
  </head>
  <body>
    <iframe src="https://www.specialist.ru" width="500" height="300"></iframe>
    <iframe src="iframe.html" width="500" height="300"></iframe>
  </body>
</html>
```

<iframe> и ссылки

Взаимодействие ссылок с элементами **iframe** в веб-разработке - это важная тема. В основном, **iframe** используется для внедрения внешних HTML-страниц на вашу веб-страницу

Когда вы создаете ссылку внутри **iframe**, по умолчанию, любой клик по этой ссылке будет загружать новую страницу внутри этого же **iframe**. Однако, вы можете изменить это поведение, используя атрибут **target** в теге **a** вашей ссылки. Например, если вы хотите, чтобы ссылка открывалась в новой вкладке браузера, вы можете использовать **target="_blank"**

Если вы хотите, чтобы ссылка открывалась в родительском окне (то есть в окне, которое содержит **iframe**), вы можете использовать **target="_parent".** Если вы хотите, чтобы ссылка открывалась в самом верхнем окне (то есть в окне, которое содержит все остальные окна), вы можете использовать **target="_top"**

<iframe> и ссылки

Index.html

```
<body>
  <iframe src="iframe.html" width="300" height="200">

  </iframe>
</body>
```

iframe.html

```
<body>
  <h1>Document</h1>
  <a href="https://www.specialist.ru" target="_blank">Ссылка</a>
</body>
```

HTML - Формы

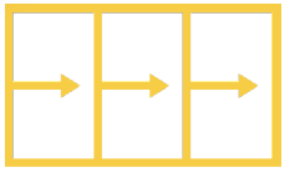
HTML-формы - это мощный инструмент для взаимодействия с пользователями на веб-страницах. Они представляют собой раздел документа, который позволяет пользователям вводить информацию для последующей обработки системой

Форма в HTML задаётся с помощью элемента <form>. Внутри этого элемента размещаются различные элементы управления (controls), такие как текстовые поля, кнопки, переключатели и другие

```
<body>
  <form action="/submit_form" method="post">
    <label for="name">Имя:</label><br />
    <input type="text" id="name" name="name" /><br />
    <input type="submit" value="Отправить" />
  </form>
</body>
```

Три способа построить макет в CSS

1



FLOAT LAYOUTS

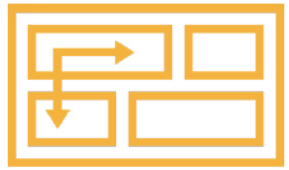
Старый способ создания макетов, использует CSS свойство **float**. Еще встречается, но быстро устаревает.

2

FLEXBOX

Современный способ выравнивания элементов в один ряд. Идеален для **макетов компонентов**.

3



CSS GRID

Используется для постройки сетки более одного ряда. Идеален для **макетов сайтов** и **сложных компонентов**.

Что такое float?

FLOAT

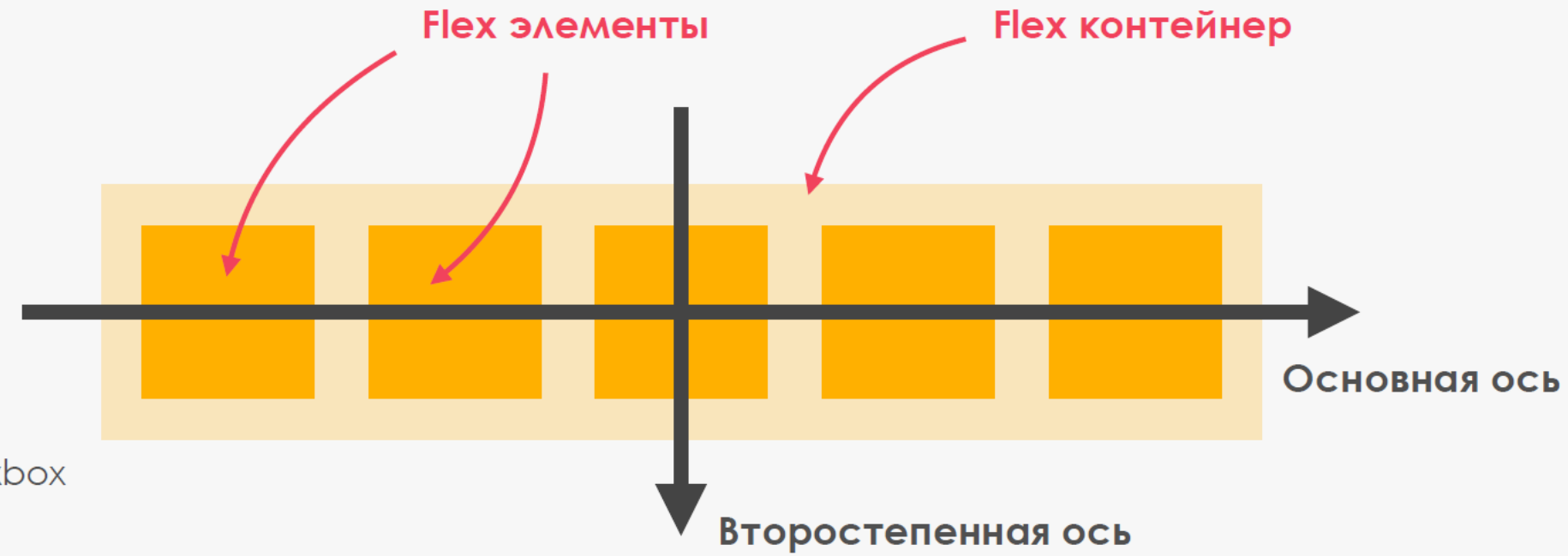
- Float - это простое **CSS свойство**, позволяющее смещать элементы вправо или влево от основного контента
- Во времена отсутствия современных инструментов размещения размещения это был основной вариант
- Применение свойства float совместно с табличной верстkой позволяли добиться разного расположения элементов на странице
- Сейчас это свойство не используется для переопределения положения в макете, но отдельные элементы с его помощью можно двигать
- Это свойство важно знать для поддержки старого кода

Что такое flexbox?

FLEXBOX

- Flexbox это набор связанных **CSS свойств**
- Основная идея flexbox в том, чтобы **распределить** место в родительском элементе между дочерними
- Flexbox **упрощает выравнивание** дочерних элементов в родительском как горизонтально, так и вертикально
- Flexbox решает проблему **вертикального выравнивания** элементов и создания блоков **одинаковой высоты**
- Flexbox идеален для **замены float**, позволяет нам писать более чистый и компактный код

Терминология flexbox



Flexbox

```
display: flex
```

Flex контейнер

1. `gap: 0 | <length>`
   - Создает **промежутки** между элементами
2. `justify-content: flex-start | flex-end | center | space-between | space-around | space-evenly`
   - Выравнивает элементов по **основной** оси
3. `align-items: stretch | flex-start | flex-end | center | baseline`
   - Выравнивает по **второстепенной** оси
4. `flex-direction: row | row-reverse | column | column-reverse`
   - **Выбирает** главную ось
5. `flex-wrap: nowrap | wrap | wrap-reverse`
   - Регулирует перенос элементов на **новую** строку
6. `align-content: stretch | flex-start | flex-end | center | space-between | space-around`
   - Выравнивает ряды, только если **более одного** ряда

Flex элементы

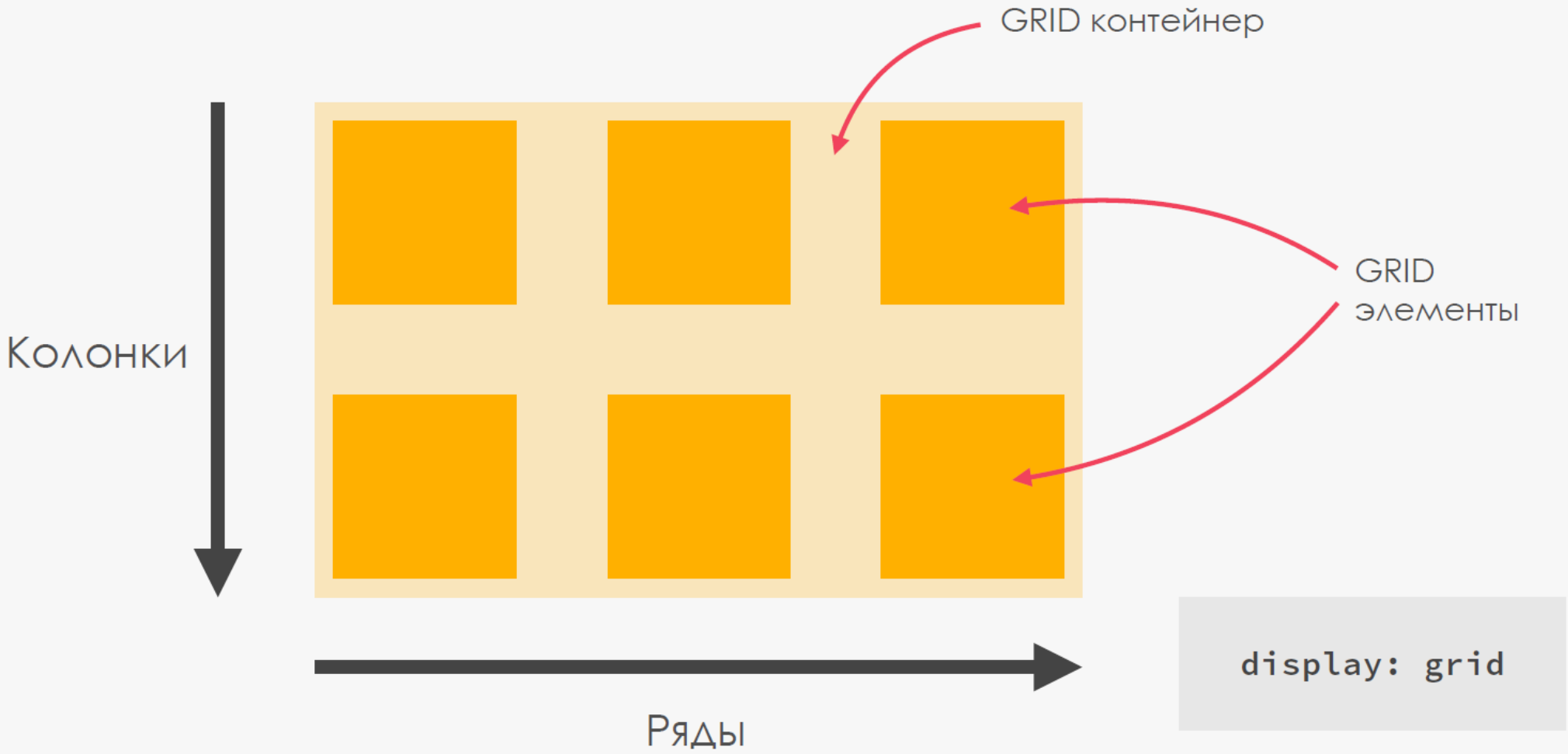
1. `align-self: auto | stretch | flex-start | flex-end | center | baseline`
   - Перезаписывает **align-items** элемента
2. `flex-grow: 0 | <integer>`
   - Позволяет элементу **увеличиваться** (0 или 1)
3. `flex-shrink: 1 | <integer>`
   - Позволяет элементу **уменьшаться** (0 или 1)
4. `flex-basis: auto | <length>`
   - Задает **ширину** элемента (вместо width)
5. `flex: 0 1 auto | <int> <int> <len>`
   - Короткая запись **flex-grow, flex-shrink, flex-basis**
6. `order: 0 | <integer>`
   - Контролирует **порядок** элементов

Что такое CSS GRID?

CSS GRID

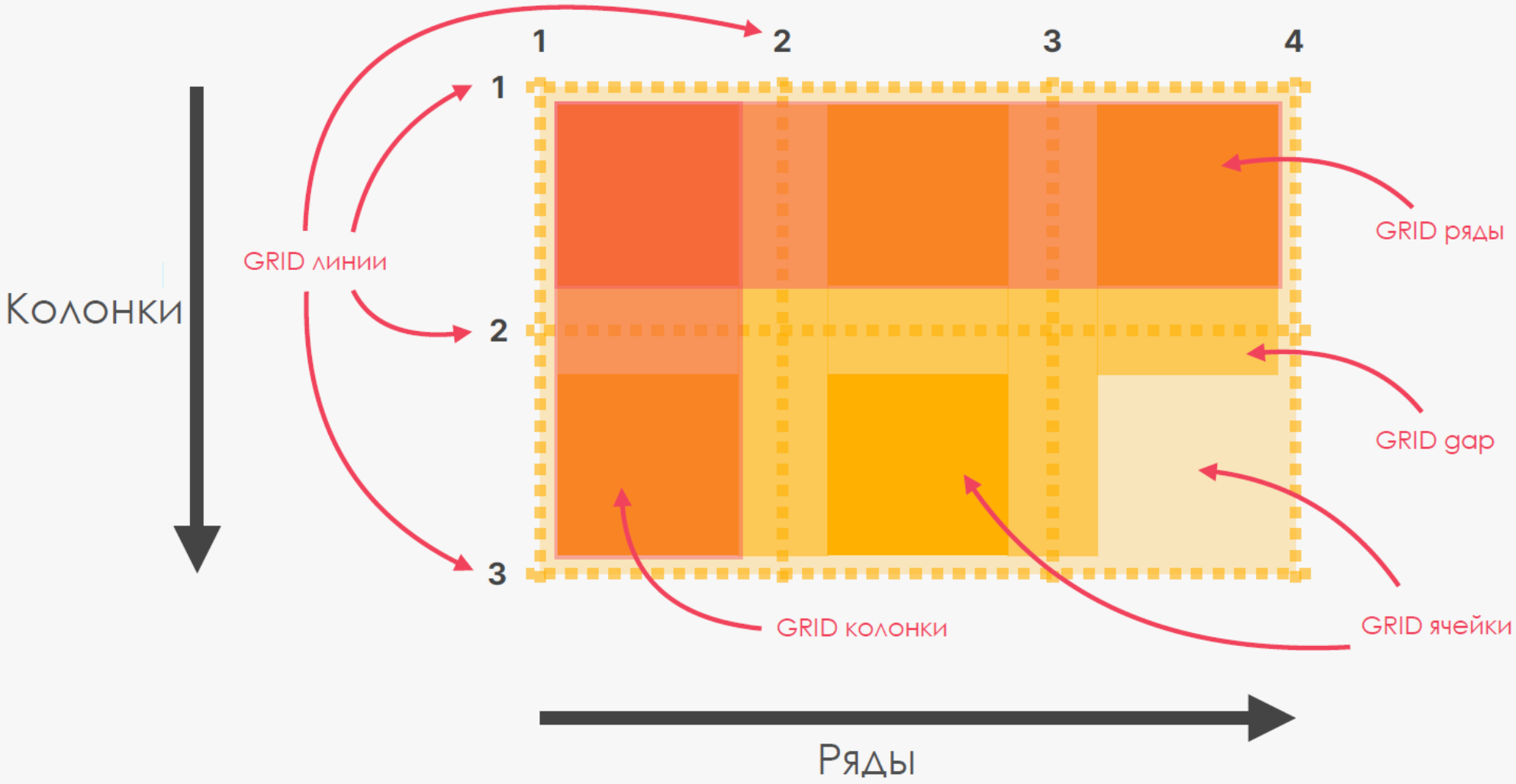
- CSS GRID - это набор связанных **CSS свойств** для постройки сложных макетов
- Основная идея CSS GRID в том, чтобы **разделить контейнер на ряды и колонки** и заполнить дочерними элементами
- CSS GRID позволяет писать **меньше вложенного HTML** кода и легче читать CSS код
- CSS GRID **не заменяет** Flexbox, они работают вместе. Нужен один ряд элементов используй Flexbox, несколько - CSS GRID

Базовая терминология CSS GRID




```
display: grid
```

Базовая терминология CSS GRID



GRID контейнер

1

```
grid-template-rows: <track size>*
grid-template-columns: <track size>*
```

👉 Задает ряды и колонки. Каждому юниту задается свое значение

2

```
row-gap: 0 | <length>
column-gap: 0 | <length>
```

] `gap: 0 | <length>`

👉 Создает пустое пространство

3

```
justify-items: stretch | start | center | end
align-items: stretch | start | center | end
```

👉 Выравнивает элементы в рядах/колонках

4

```
justify-content: start | start | center | end | ...
align-content: start | start | center | end | ...
```

👉 Выравнивает сетку внутри контейнера, если контейнер больше сетки

GRID элементы

1

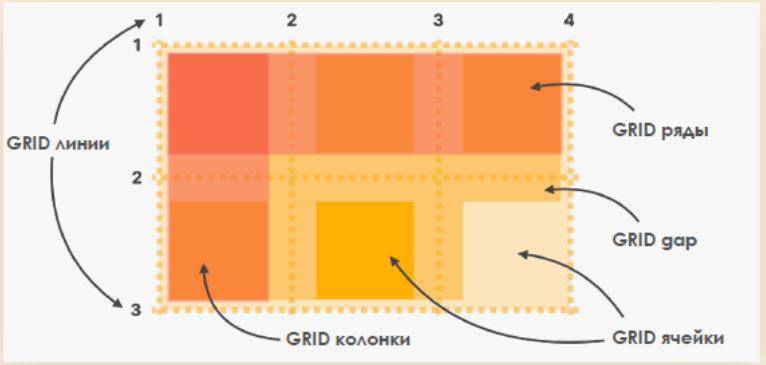
```
grid-column: <start line> / <end line> | span <number>
grid-row: <start line> / <end line> | span <number>
```

👉 Задает место элементу в ячейке сетки. Span регулирует количество ячеек

2

```
justify-self: stretch | start | center | end
align-self: stretch | start | center | end
```

👉 Перезаписывает justify-items/align-items элемента



Спасибо
за внимание!

Ваши вопросы...

Учебный центр «СПЕЦИАЛИСТ» –

Ваш путь к успеху

info@specialist.ru

+7 (495) 232-32-16