

# Модуль - Верстка

---

9/10 занятие

# АДАПТИВНОСТЬ

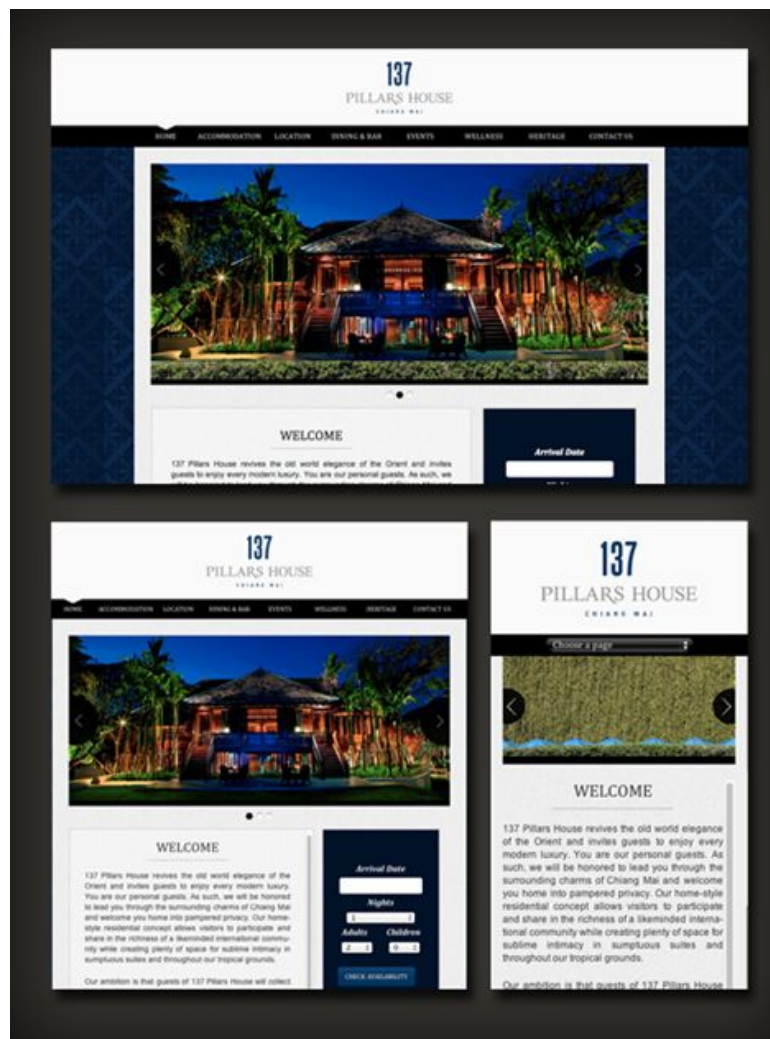
---

**Адаптивность**(responsive) - Способность сайта или приложения корректно отображать контент на различных устройствах, таких как:

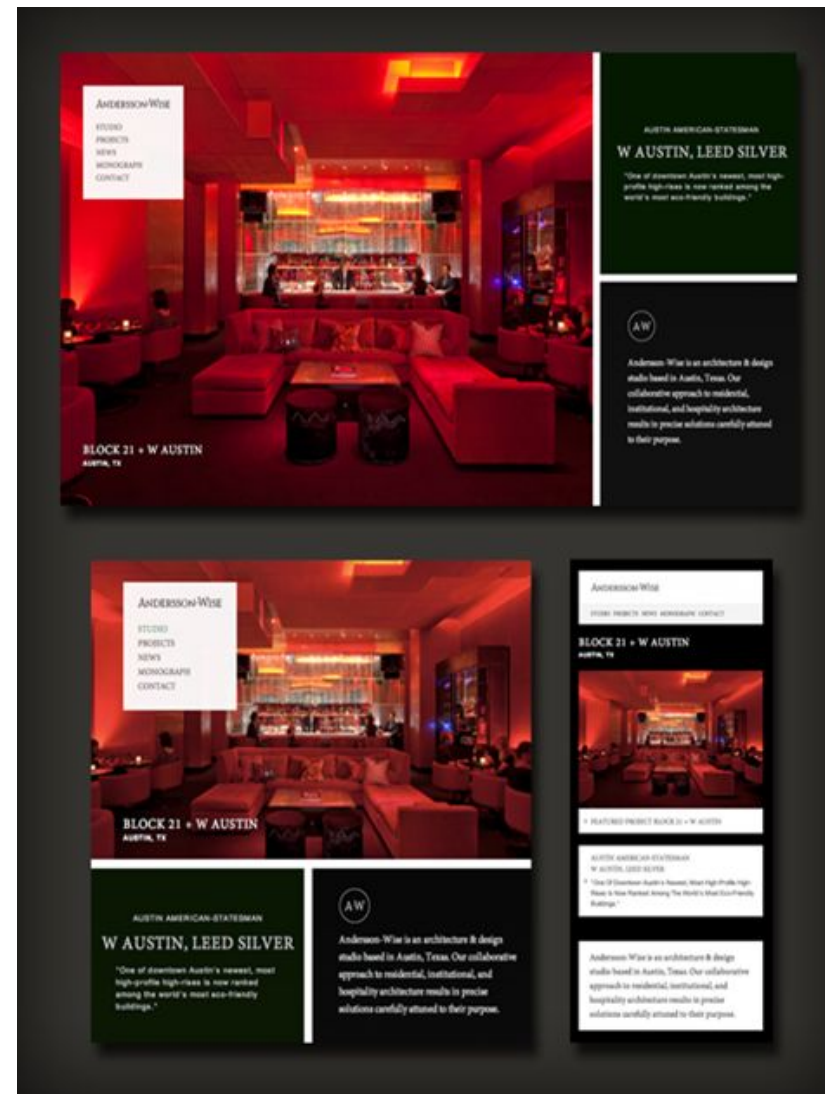
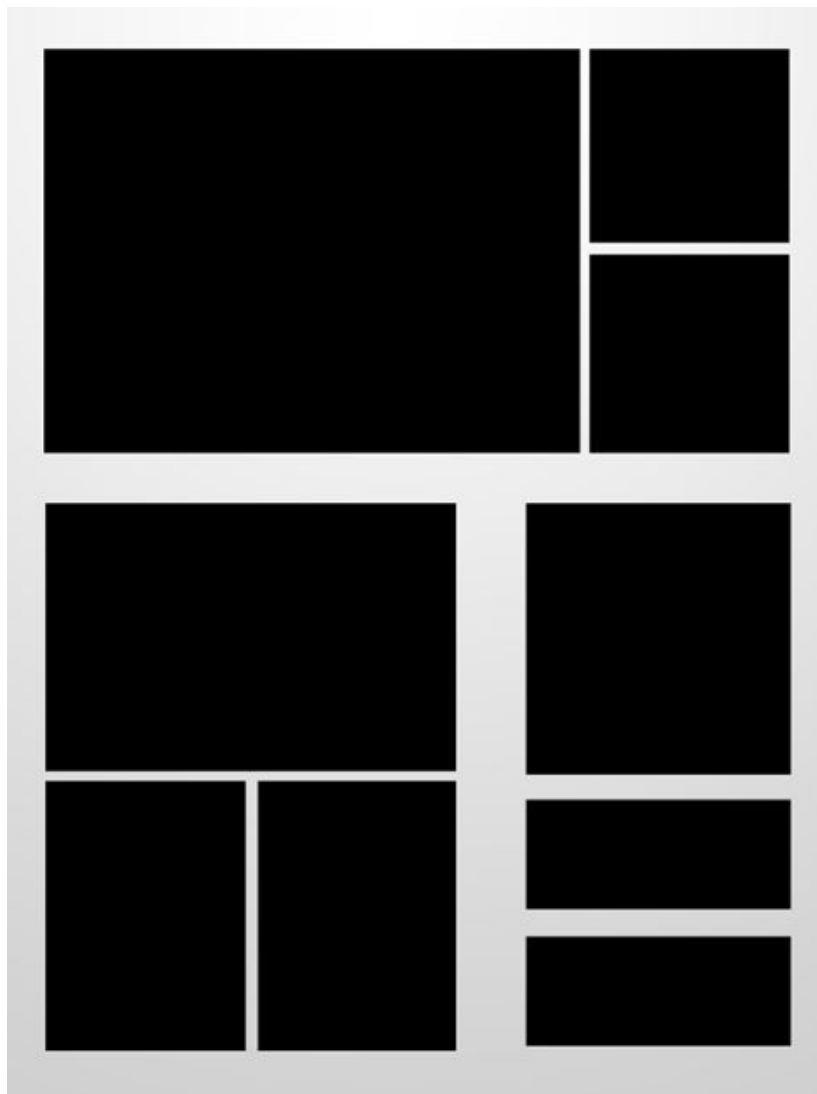
- Мониторы.
- Телевизоры;
- Планшеты;
- Смартфоны;

Концепция заключается в использовании “*фильтров*” для css правил, тем самым, задействуя или игнорируя их, в зависимости от того, на каком устройстве запущен браузер

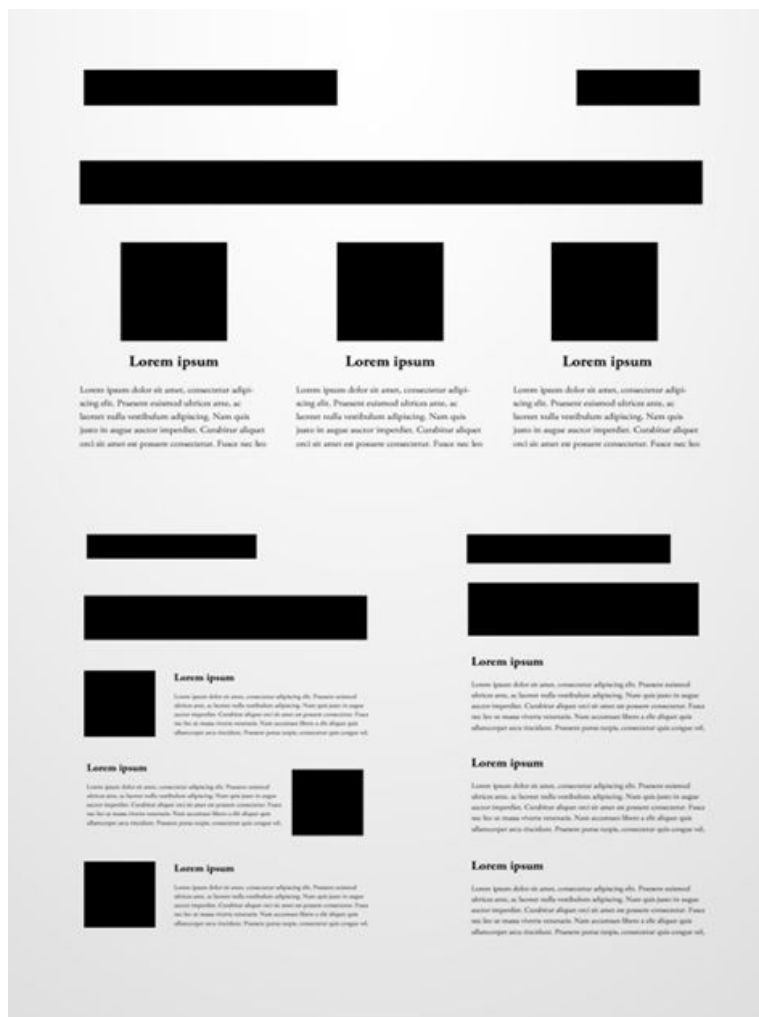
# ПРОЯВЛЕНИЕ АДАПТИВНОСТИ



# ОТЗЫВЧИВАЯ КОМПОЗИЦИЯ



# ИЗМЕНЕНИЕ ВИДИМОСТИ И СТИЛЕЙ



# МАСШТАБ ОБЛАСТИ ПРОСМОТРА

---

Мета-тег **viewport** определяет то, как веб-страница отображается на мобильном устройстве.

Если он не указан, ширина страницы считается равной стандартному значению для ПК, и она уменьшается, чтобы поместиться на экране.

```
<head>
  ...
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  ...
</head>
```

# ФИЛЬТРАЦИЯ СТИЛЕЙ

---

Для определения того, какие правила будут задействованы, а какие нет, используются **медиазапросы**(**media-queries**)

Они позволяют *изменять стили* на основании *характеристик устройства*, связанных с отображением контента, включая *тип, ширину, высоту, ориентацию* и даже *разрешение экрана*.

Чаще всего используется фильтр по максимальной(**max-width: 991px**) или минимальной(**min-width: 768px**) ширине или диапазон ширин.

320px

768px

1024px

1920px

4096px



# Как использовать

---

1. Вращеры внутри CSS: **@media <условие> { ... css-правила ... }**
2. Внутри тега **<link>**

Условие может содержать:

- типы носителей (all, screen, print, ...);
- логические операторы (and, not, only);
- медиа-функции(включающие свойства css).

CSS:

```
@media only screen and (min-width: 320px) and (max-width: 768px) {  
  .col {  
    width: 50%;  
  }  
}
```

HTML:

```
<link href="mobile.css" rel="stylesheet" media="screen and (max-width: 600px)" />
```



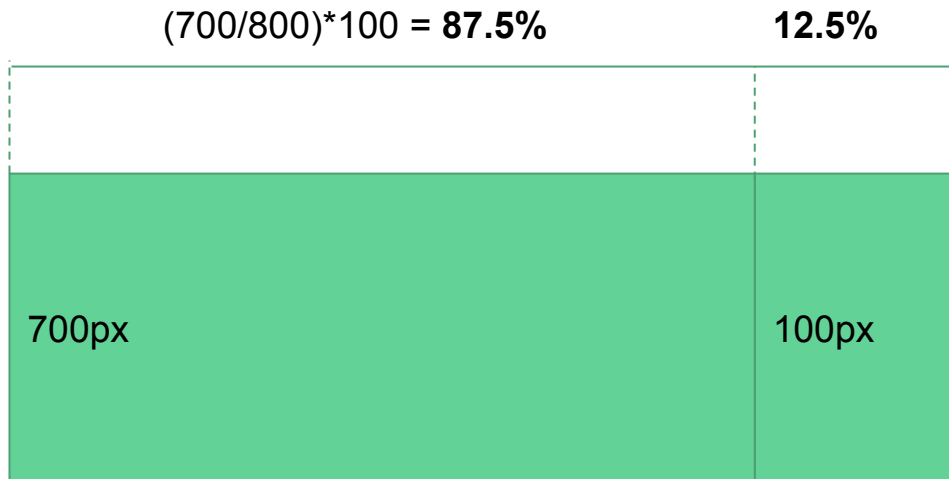
# ВЫЧИСЛЕНИЕ ПРОПОРЦИЙ

---

$\% = (\text{element-size} / \text{parent-size}) * 100$  , где

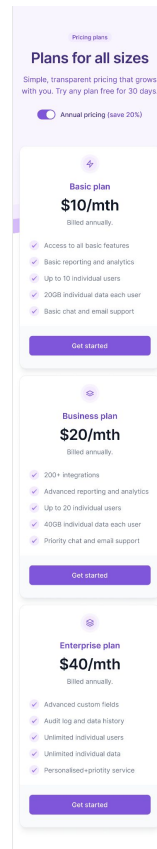
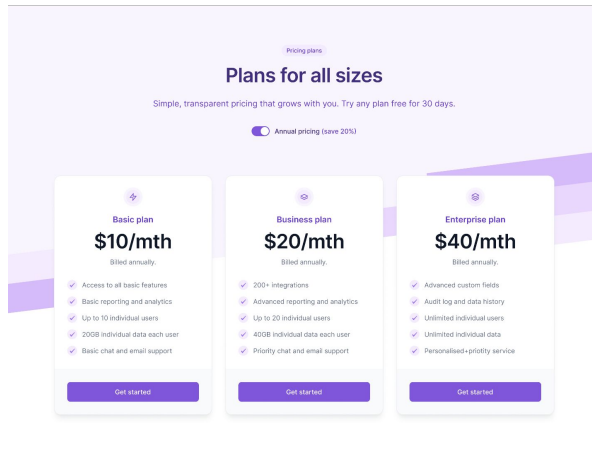
- $\text{parent-size}$  – размер родительского контейнера в макете
- $\text{element-size}$  – размер элемента

*Единицы измерения  $\text{parent-size}$  и  $\text{element-size}$  должны совпадать!*



# Задание 1

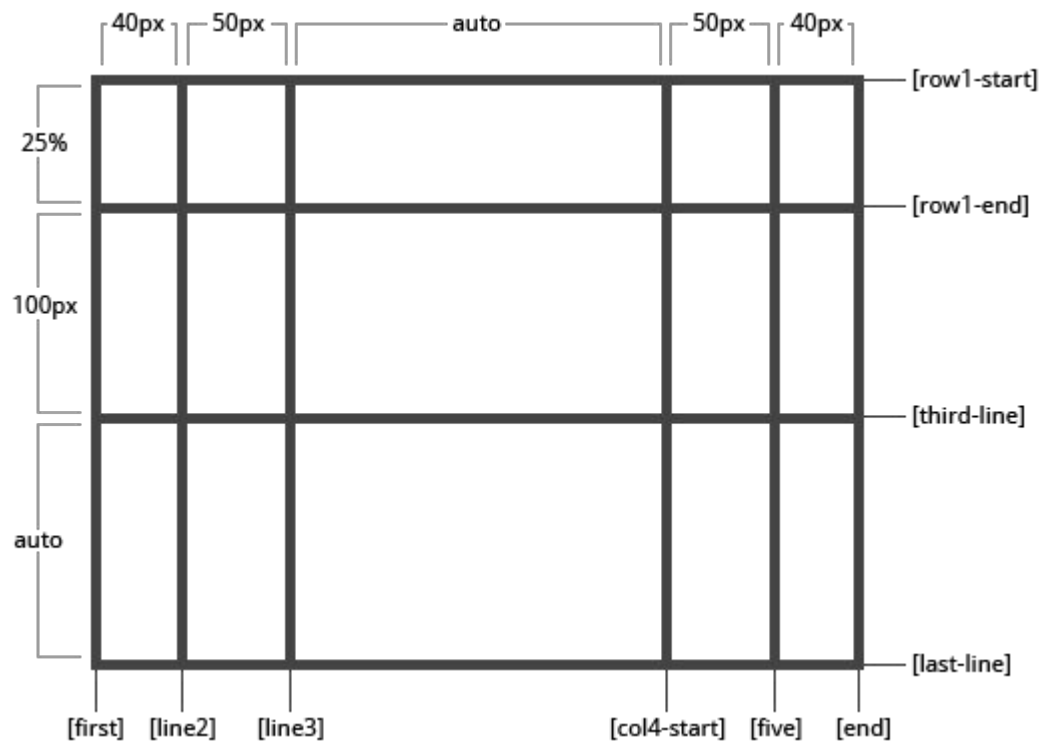
В Landing-ехартпле скрыть все секции, кроме указанных на скриншоте и добавить адаптивные стили через @media-запросы в файл mobile.css



# GRID LAYOUT

---

# Сетка



# Где Grid, а где Flex

---

**Flexbox** - для макета в одном измерении. *Либо в строке , либо в столбце*

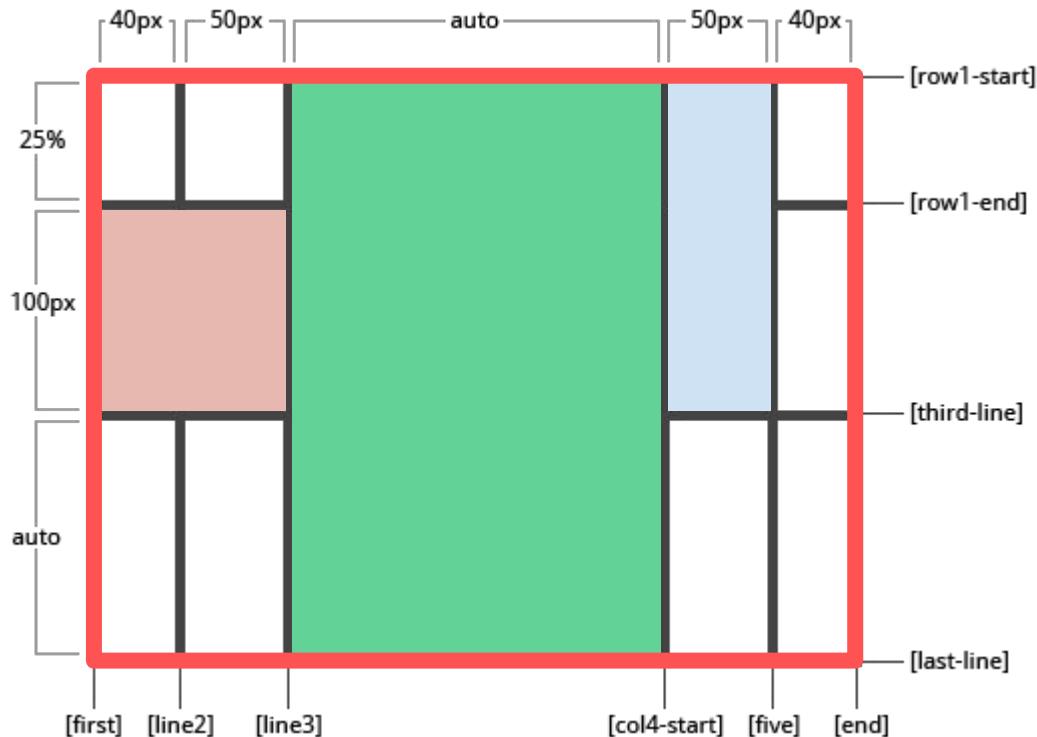
**Grid** - для двухмерной компоновки строк и столбцов одновременно

Оба в коде представляют 2 уровня(контейнер и элементы контейнера)

```
<div class="grid-container">
  <div class="item">One</div>
  <div class="item">Two</div>
  <div class="item">Three</div>
  <div class="item">Four</div>
</div>
```

```
<> <div class="flex-container">
  <div class="item">One</div>
  <div class="item">Two</div>
  <div class="item">Three</div>
  <div class="item">Four</div>
</div> <>
```

# Grid - контейнер



```
.grid-container {
```

```
  display: grid;
```

```
  grid-template-columns: 40px, 50px, 1fr, 50px, 40px;
```

```
  grid-template-rows: 25%, 100px, auto;
```

```
  grid-gap: 10px 15px;
```

```
}
```

//режим grid-контейнера

//колонки

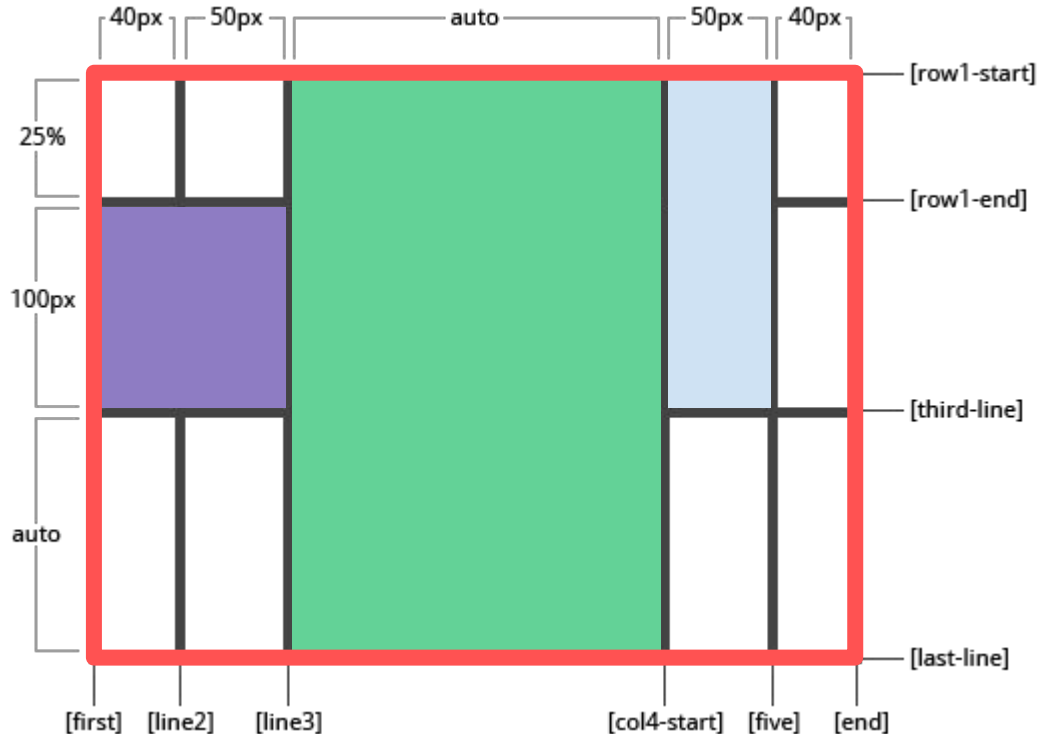
//ряды

//разделитель

# Элементы Grid - контейнера

**grid-row:**  
ряды(от - до)

**grid-column:**  
колонки(от - до)



```
.grid-item-violet {  
  grid-row: 2 / 3;  
  grid-column: 1 / 3;  
}
```

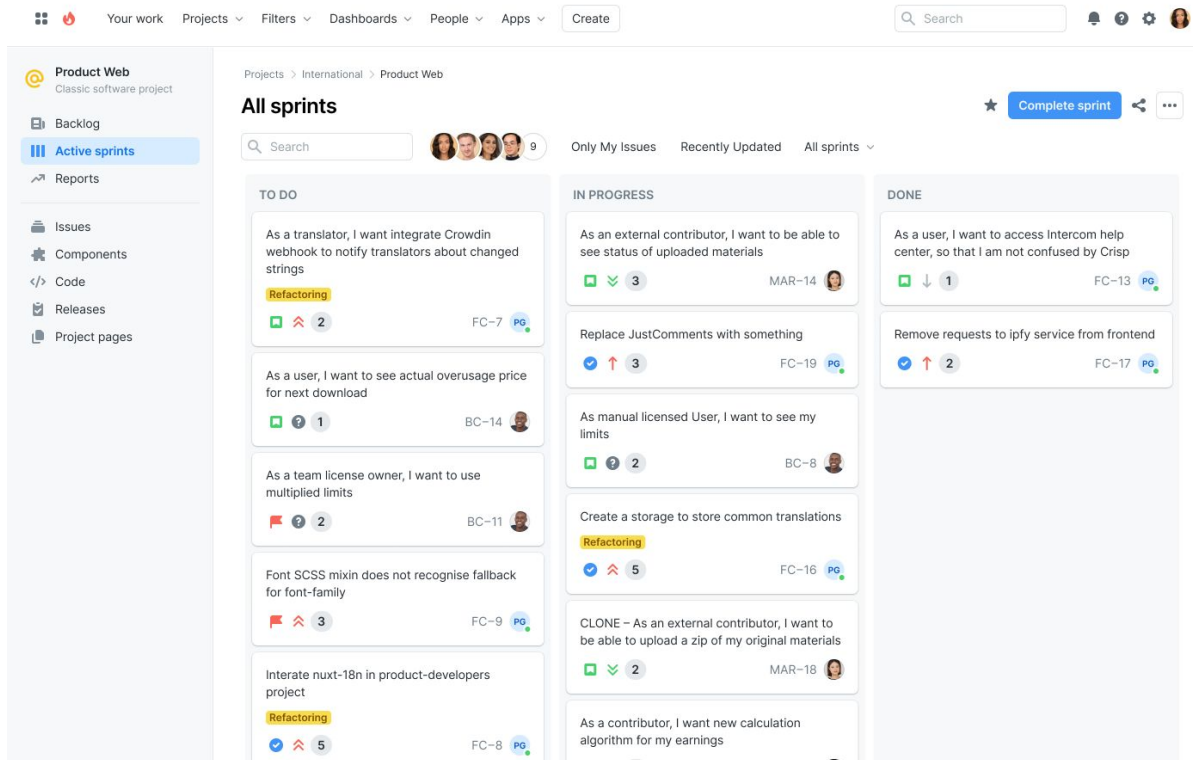
```
.grid-item-mint {  
  grid-row: 1 / 4;  
  grid-column: 3 / 4;  
}
```

```
.grid-item-blue {  
  grid-row: 1 / 3;  
  grid-column: 4 / 5;  
}
```

# Задание 2

Создать kanban-board.html, подключить css

Сверстать по фигме, используя только grid режим:





**перерыв**

# Библиотеки

---

**Bootstrap** - одна из самых популярных и функциональных css библиотек

<https://getbootstrap.com/docs/5.3/getting-started/introduction/>

**Подключение:**

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Bootstrap demo</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-T3c6Coli6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN"
crossorigin="anonymous">
  </head>
  <body>
    <h1>Hello, world!</h1>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL"
crossorigin="anonymous"></script>
  </body>
</html>
```

# Задание 3

---

1. Создать site-on-bootstrap.html
2. Подключить библиотеку bootstrap 5
3. Сверстать по эскизу используя компоненты
  - Контейнер <https://getbootstrap.com/docs/5.3/layout/containers/>
  - Меню <https://getbootstrap.com/docs/5.3/components/navbar/>
  - Колонки <https://getbootstrap.com/docs/5.3/layout/columns/>
  - Карточка с картинкой <https://getbootstrap.com/docs/5.3/components/card/>
  - Формы <https://getbootstrap.com/docs/5.3/forms/overview/>

Navbar

Home

Link

Dropdown ▾

Disabled

Search

Search

Disabled fieldset example

Disabled input

Disabled input

Disabled select menu

Disabled select ▾

☐ Can't check this

Submit

Image cap

Card title

Some quick example text to build on the card title and make up the bulk of the card's content.

Go somewhere

# Практика: сборка проекта

---

Удобный и простой гайд по настройке сборщика vite + bootstrap

<https://getbootstrap.com/docs/5.3/getting-started/vite/>

**Конец занятия**