

Статическая верстка

Модуль 1 / 5

Технологии

HTML



CSS



JS



Первоначальный сброс настроек

Селектор * - означает, что правила будут применены к всем элементам документа.

font-family: sans-serif; - делает шрифт без засечек

margin: 0; - сбрасывает внешние отступы

padding: 0; - сбрасывает внутренние отступы

box-sizing: border-box; - задает правильную калькуляцию размера бокса:

<https://developer.mozilla.org/ru/docs/Web/CSS/box-sizing>

```
* {  
    font-family: sans-serif;  
    margin: 0;  
    padding: 0;  
    box-sizing: border-box;  
}
```

Ширина и высота

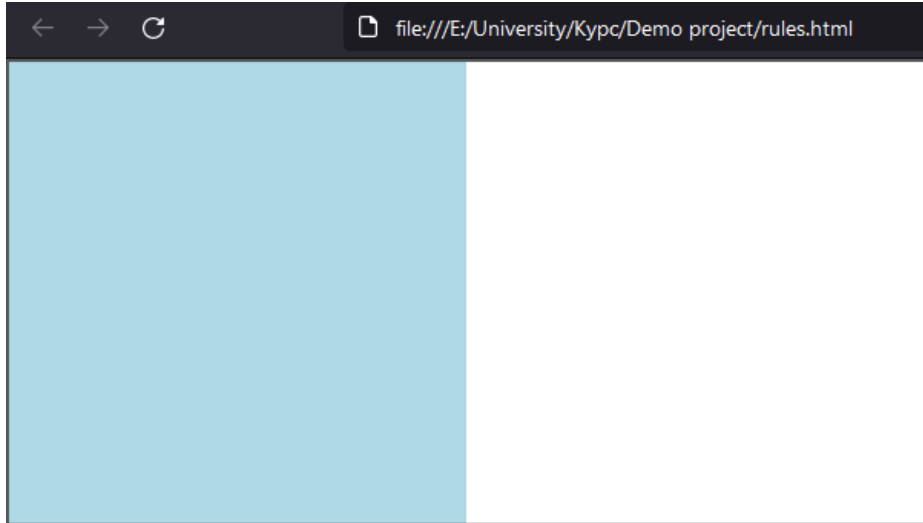
В пикселях – абсолютное значение



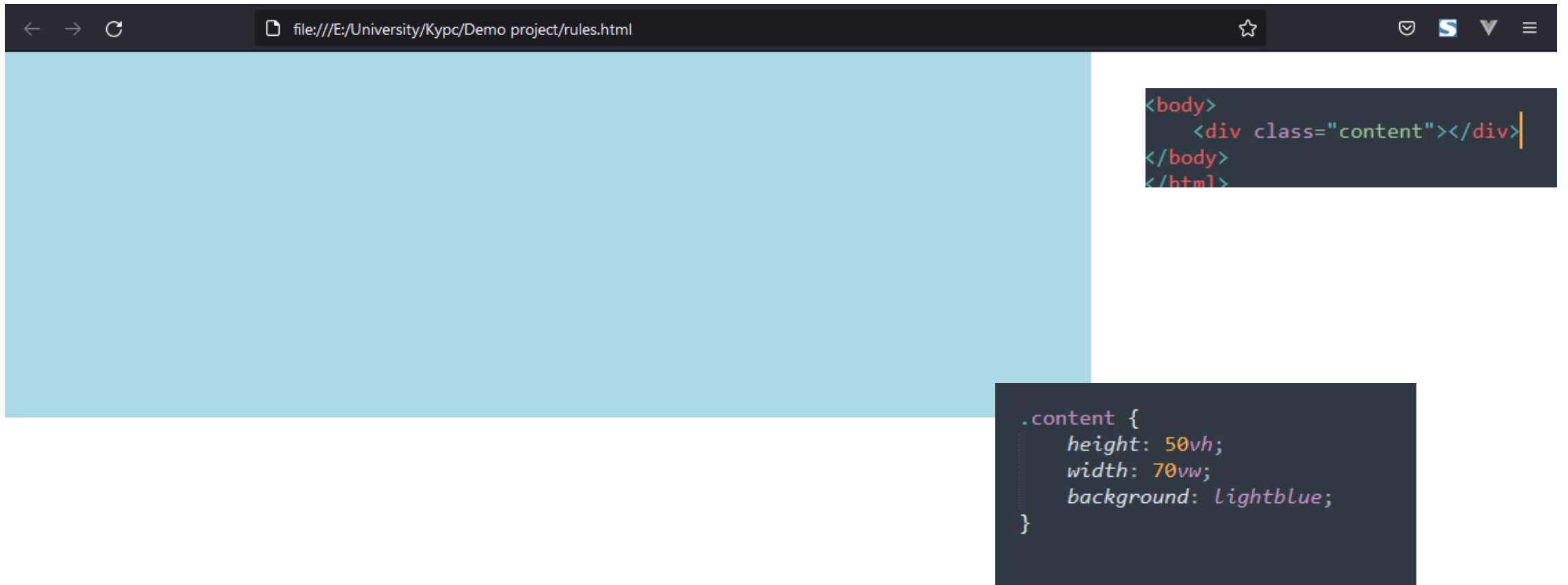
В % - относительно родителя!

```
<body>
  <div class="wrapper">
    <div class="content"></div>
  </div>
</body>
```

```
.wrapper {
  border: 2px solid #565656;
  width: 600px;
}
.content {
  height: 300px;
  width: 50%;
  background: lightblue;
}
```



Ширина в vw, высота в vh – от экрана



Свойство display

Управляет тем, как элементы будут расположены на странице

<https://developer.mozilla.org/ru/docs/Web/CSS/display>

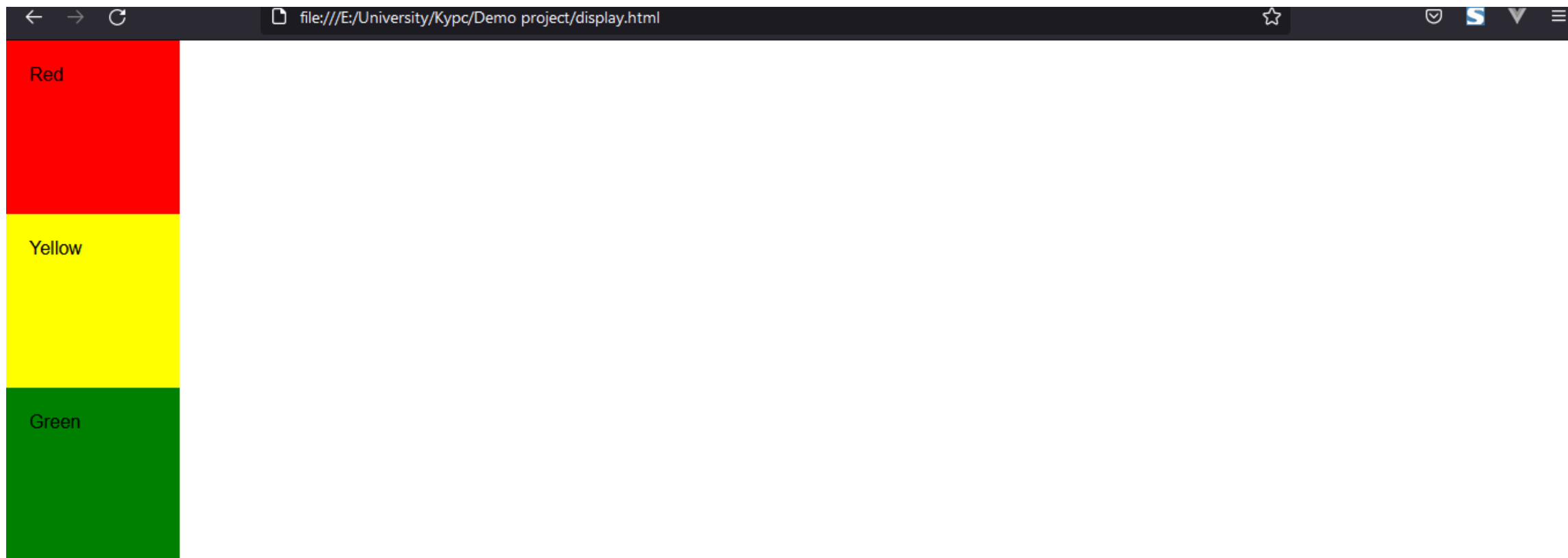
Значений довольно много, на практике чаще всего используется block, flex, inline, grid, inline-flex, inline-grid. Кстати, строки таблиц имеют display table-row, а ячейки table-cell.

Для начала сделаем такой макет

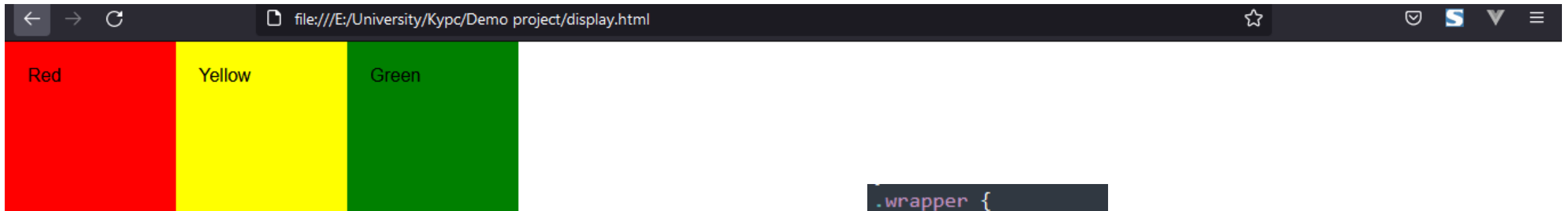
У нас блок – обертка, высотой сделали во весь экран, чтобы показать выравнивание по вертикали. В нем 3 блока 150*150, красный, желтый и зеленый
Эти 3 блока имеют внутренний отступ 20 пикселей

```
<style>
  * {
    font-family: sans-serif;
    margin: 0;
    padding: 0;
    box-sizing: border-box;
  }
  .wrapper {
    height: 100vh;
  }
  .block-item {
    width: 150px;
    height: 150px;
    padding: 20px;
  }
  .red {
    background: red;
  }
  .yellow {
    background: yellow;
  }
  .green {
    background: green;
  }
</style>
</head>
<body>
  <div class="wrapper">
    <div class="block-item red">Red</div>
    <div class="block-item yellow">Yellow</div>
    <div class="block-item green">Green</div>
  </div>
</body>
```

Block – по умолчанию



Flex



```
.wrapper {  
  height: 100vh;  
  display: flex;  
}
```

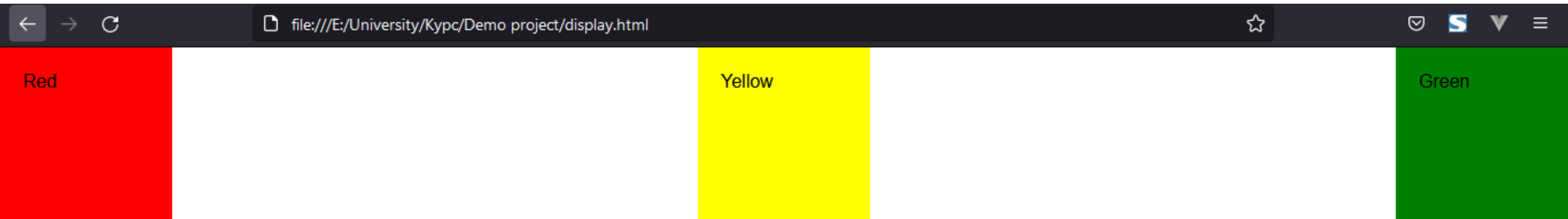
Свойство justify-content



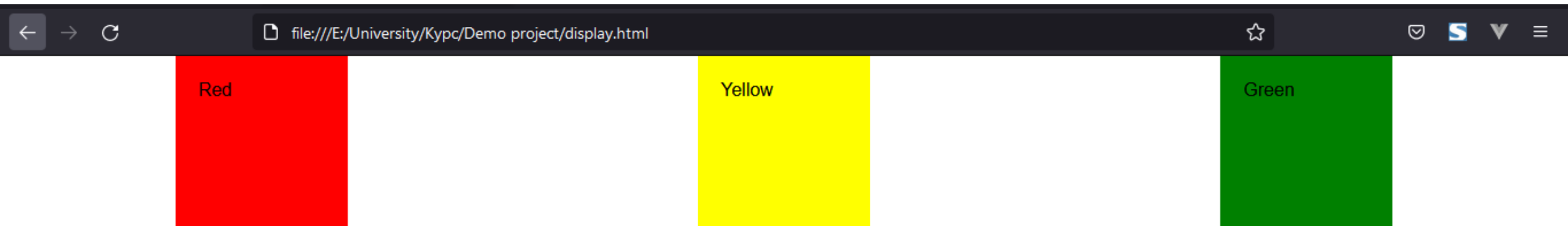
```
.wrapper {  
  height: 100vh;  
  display: flex;  
  justify-content: center;  
}
```

Свойство justify-content

```
.wrapper {  
  height: 100vh;  
  display: flex;  
  justify-content: space-between;  
}  
block-item {
```

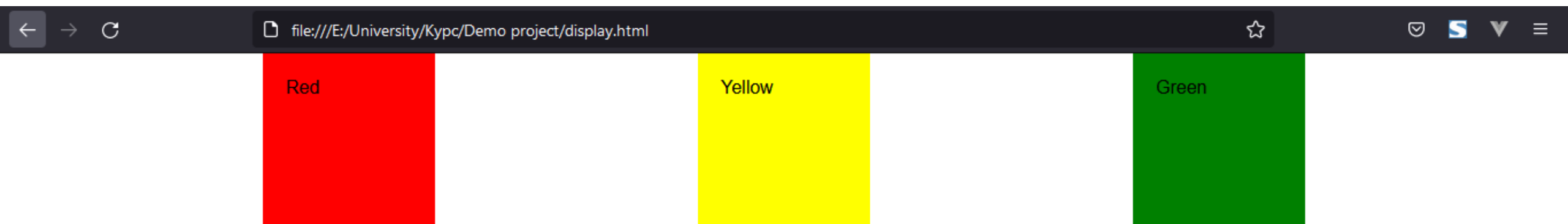


Свойство justify-content



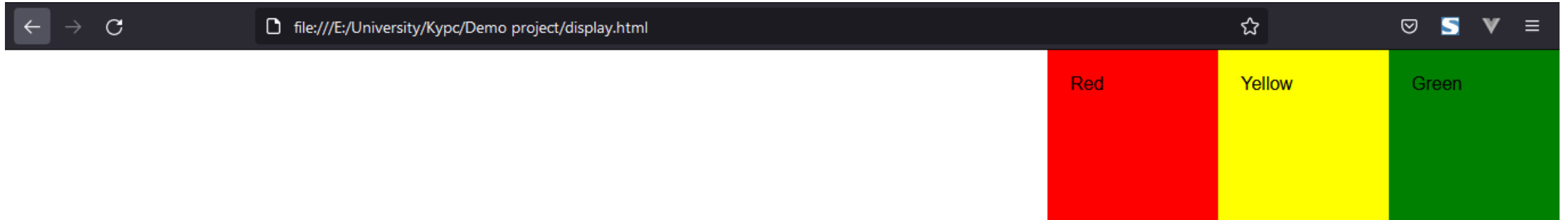
```
.wrapper {  
  height: 100vh;  
  display: flex;  
  justify-content: space-around;  
}
```

Свойство justify-content



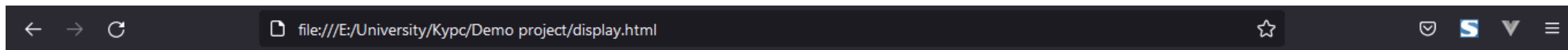
```
.wrapper {  
  height: 100vh;  
  display: flex;  
  justify-content: space-around;  
}  
.block-item {
```

Свойство justify-content



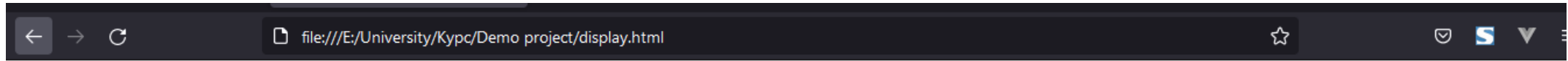
```
.wrapper {  
  height: 100vh;  
  display: flex;  
  justify-content: flex-end;  
}  
block-item {
```


Свойство align-items



```
.wrapper {  
  height: 100vh;  
  display: flex;  
  align-items: center;  
}
```

Свойство align-items

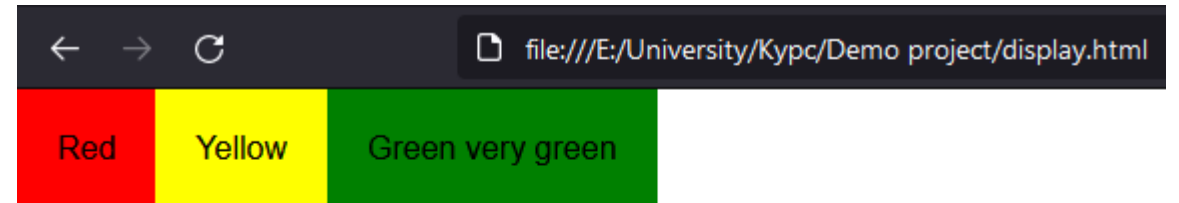
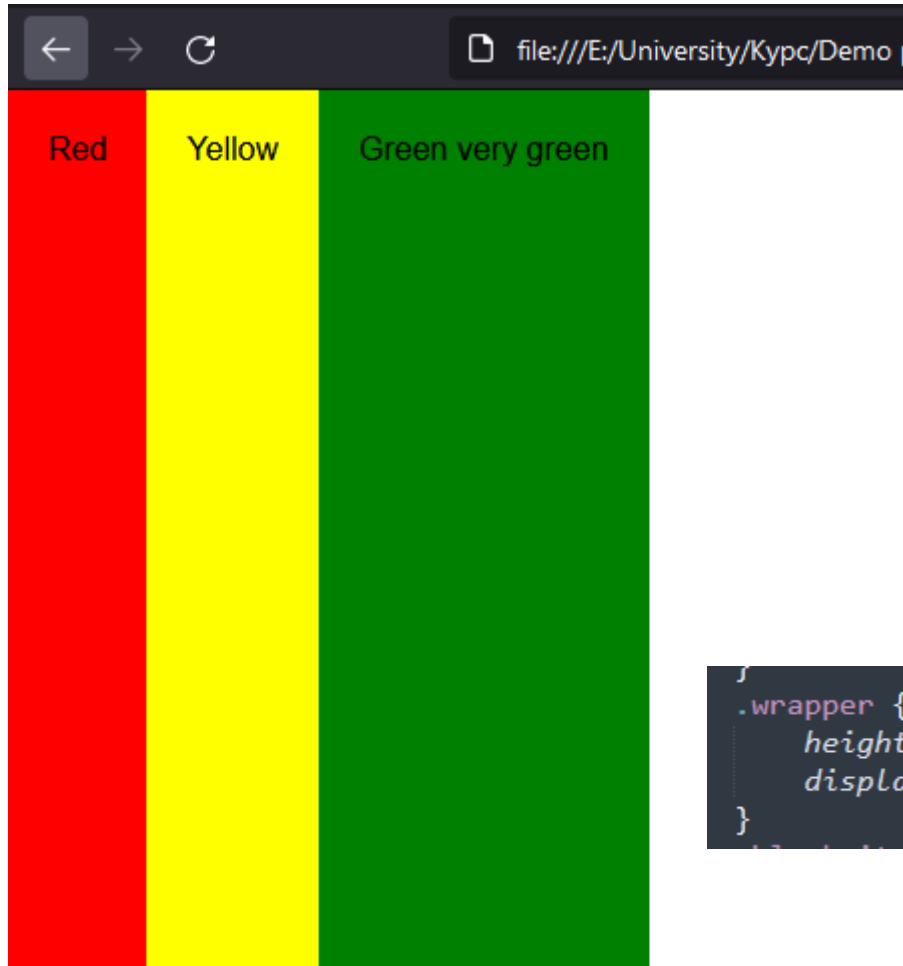


```
.wrapper {  
  height: 100vh;  
  display: flex;  
  align-items: flex-end;  
}
```



Если высота блоков явно не прописана, при значении align-items: stretch (а оно по умолчанию), блоки будут растянуты на всю высоту родителя

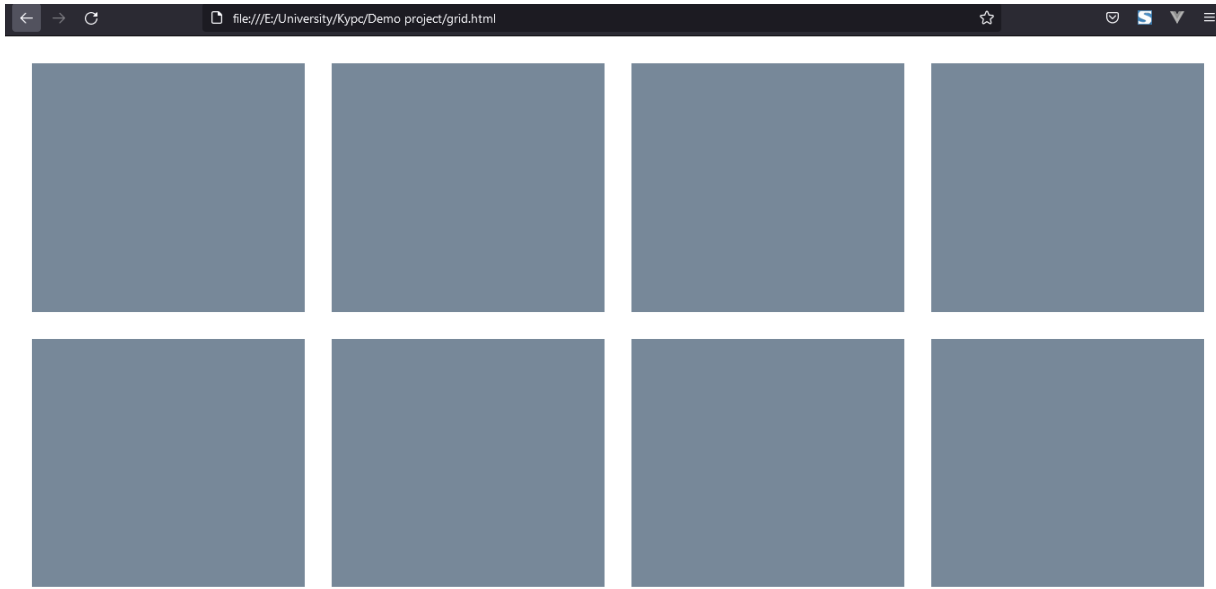
Если ширина явно не задана, ширина будет задана автоматически по содержимому контента



```
.wrapper {  
  height: 100vh;  
  display: flex;  
  align-items: flex-start;  
}
```

```
.wrapper {  
  height: 100vh;  
  display: flex;  
}
```

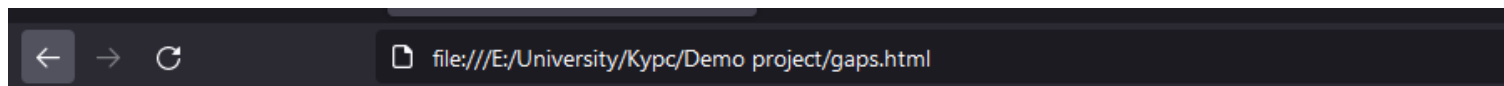
Grid



https://developer.mozilla.org/ru/docs/Web/CSS/CSS_Grid_Layout/Basic_Concepts_of_Grid_Layout

[illegible]

Отступы



Hello from .container!

```
* {  
  font-family: sans-serif;  
  margin: 0;  
  padding: 0;  
  box-sizing: border-box;  
}  
.container {  
  background: lightblue;  
  margin: 50px;  
  padding: 40px;  
}  
</style>  
</head>  
<body>  
  <div class="container">  
    Hello from .container!  
  </div>  
</body>
```

Отступы

Margin – отступ от внешнего контента

Padding – отступ от внутреннего

Можно задавать для всех сторон отдельно, показано на примере margin:

```
margin-top: 20px;  
margin-right: 40px;  
margin-bottom: 80px;  
margin-left: 50px;  
  
margin: 20px 40px 80px 50px;
```


```
margin-top: 20px;  
margin-right: 40px;  
margin-bottom: 20px;  
margin-left: 40px;  
  
margin: 20px 40px;
```

Цвет текста - color

Hello from .container!

```
* {  
    font-family: sans-serif;  
    margin: 0;  
    padding: 0;  
    box-sizing: border-box;  
}  
  
.container {  
    margin: 50px;  
    width: 400px;  
    height: 300px;  
    padding: 40px;  
    color: darkblue;  
}  
  
</style>  
</head>  
<body>  
    <div class="container">  
        Hello from .container!  
    </div>  
</body>
```

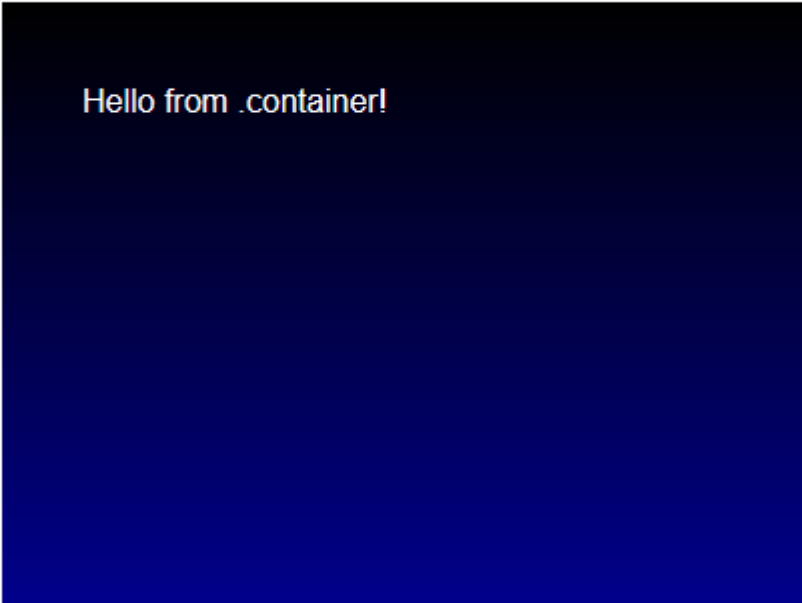
Фон – сплошной цвет



Hello from .container!

```
.container {  
  margin: 50px;  
  width: 400px;  
  height: 300px;  
  padding: 40px;  
  color: white;  
  background: #565656;  
}
```

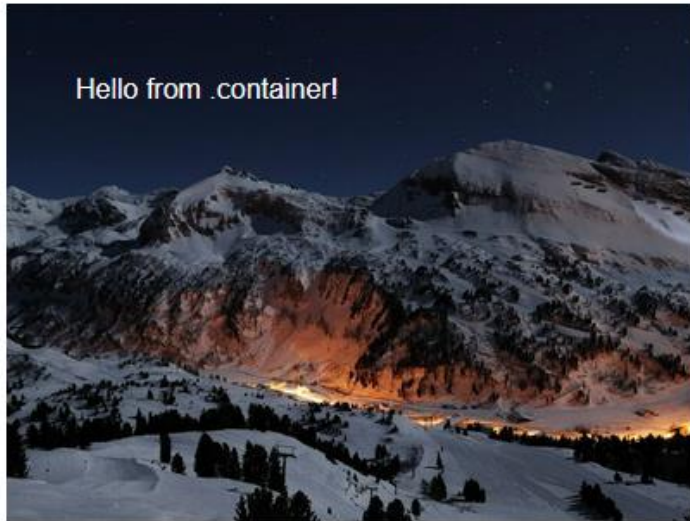

Фон - градиент



Hello from .container!

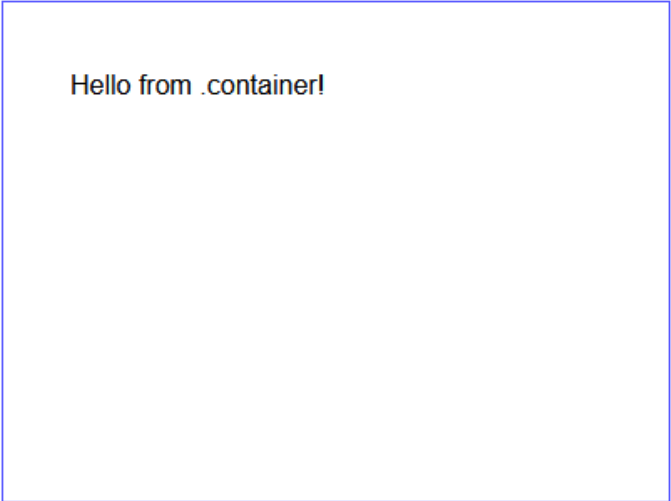
```
.container {  
  margin: 50px;  
  width: 400px;  
  height: 300px;  
  padding: 40px;  
  color: white;  
  background: linear-gradient(to bottom, black, darkblue);  
}
```

Фон-картинка



```
    }  
    .container {  
        margin: 50px;  
        width: 400px;  
        height: 300px;  
        padding: 40px;  
        color: white;  
        background: url("https://picsy.ru/images/photos/1200/1/gory-nochyu-34669.jpg");  
        background-size: cover;  
    }  
}
```


Границы



Hello from .container!

```
}  
.container {  
  margin: 50px;  
  width: 400px;  
  height: 300px;  
  padding: 40px;  
  border: 1px solid #5656FF;  
}
```

Скругление



Hello from .container!

```
.container {  
  margin: 50px;  
  width: 400px;  
  height: 300px;  
  padding: 40px;  
  border: 1px solid #5656FF;  
  border-radius: 30px;  
}
```

Тень блока

Hello from .container!

```
.container {  
  margin: 50px;  
  width: 400px;  
  height: 300px;  
  padding: 40px;  
  border: 1px solid #5656FF;  
  border-radius: 30px;  
  box-shadow: 5px 5px 20px rgba(0, 0, 0, .75);  
}
```

Тень текста

Hello from
.container!

```
.container {  
  margin: 50px;  
  width: 400px;  
  height: 300px;  
  padding: 40px;  
  border: 1px solid #5656FF;  
  border-radius: 30px;  
  text-shadow: 2px 2px 5px rgba(0, 0, 0, .75);  
  font-size: 3rem;  
}
```

Свойства шрифта – несколько примеров

```
font-size: 20px;  
font-size: 1.5rem;  
  
font-family: Arial;  
  
font-weight: bold;  
  
font-style: italic;  
  
text-align: center;  
  
text-decoration: underline;
```

Шрифты можно брать стандартные системные, либо подключать из файла, еще хорошее решение – Google Fonts

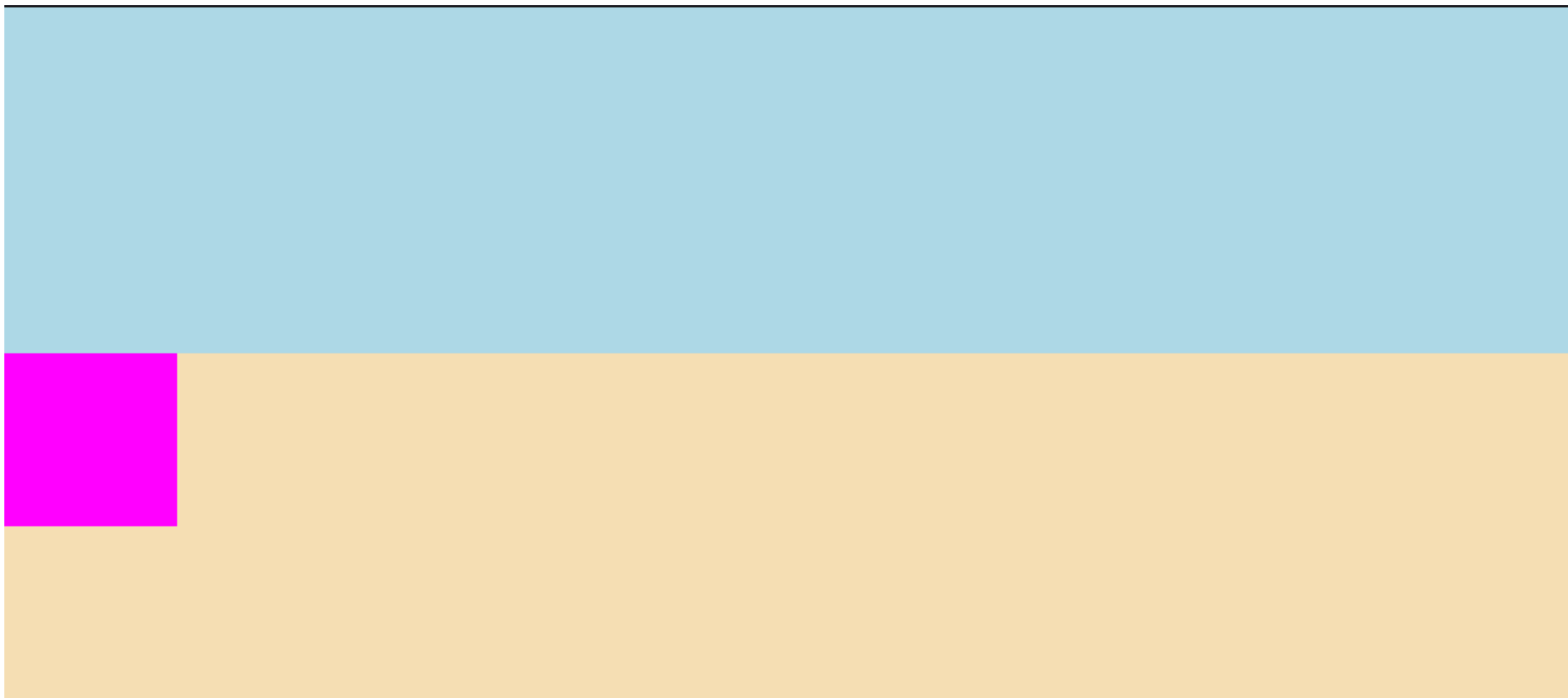
<https://fonts.google.com/>

Создадим такой макет

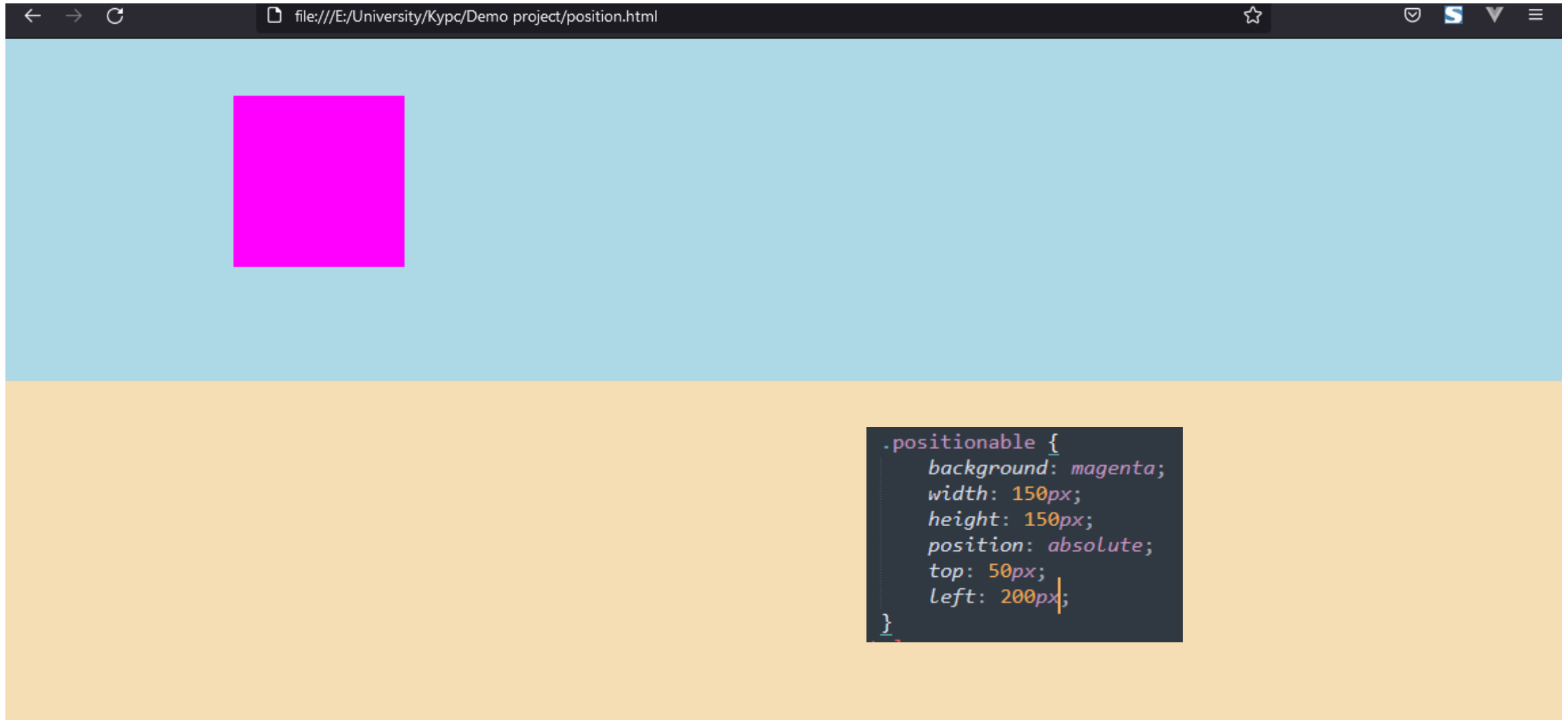
2 контейнера высотой по 300 пикселей, во втором лежит блок 150*150. Его и будем позиционировать

```
<style>
  * {
    font-family: sans-serif;
    margin: 0;
    padding: 0;
    box-sizing: border-box;
  }
  .container {
    height: 300px;
  }
  .container:first-child {
    background: lightblue;
  }
  .container:last-child {
    background: wheat;
  }
  .positionable {
    background: magenta;
    width: 150px;
    height: 150px;
  }
</style>
</head>
<body>
  <div class="container"></div>
  <div class="container">
    <div class="positionable"></div>
  </div>
</body>
```


Position – по умолчанию static



Position absolute



Position absolute внутри relative



Список всех свойств CSS

<https://schoolsw3.com/cssref/index.php>