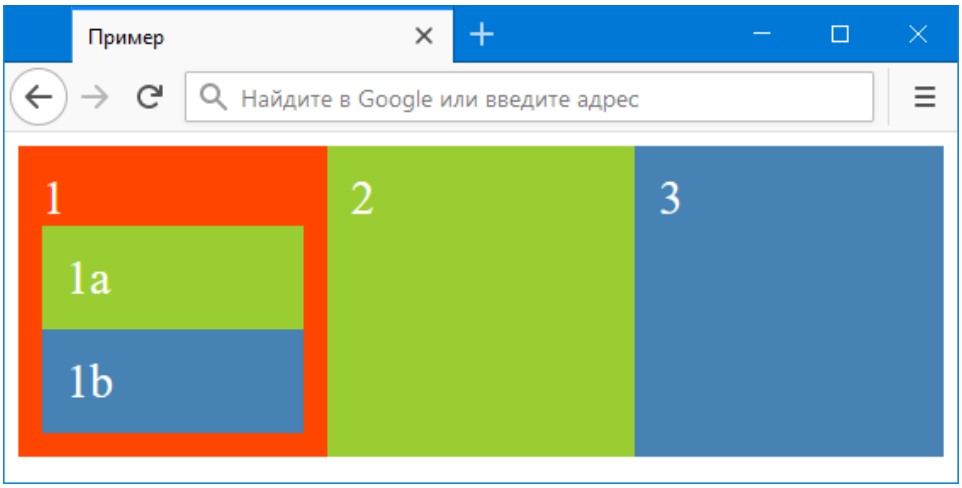
Вложенные флекс-контейнеры

Флексбокс по своей сути является моделью одномерного макета. Флекс-элементы внутри флекс-контейнера можно размещать как горизонтально, так и вертикально, но не одновременно. Если вы хотите располагать элементы в двух направлениях, вам потребуется вставить один флекс-контейнер в другой.

Вроде этого.

☐ Тёмная тема

```
<!doctype html>
<title>Пример</title>
<style>
.container {
  display: flex;
}
.red {
  background: orangered;
  display: flex;
  flex-direction: column;
}
.green {
  background: yellowgreen;
}
.blue {
  background: steelblue;
}
.container div {
  font-size: 5vw;
  padding: .5em;
  color: white;
  flex: 1;
}
</style>
<div class="container">
  <div class="red">1
    <div class="green">1a</div>
    <div class="blue">1b</div>
  </div>
  <div class="green">2</div>
  <div class="blue">3</div>
</div>
```

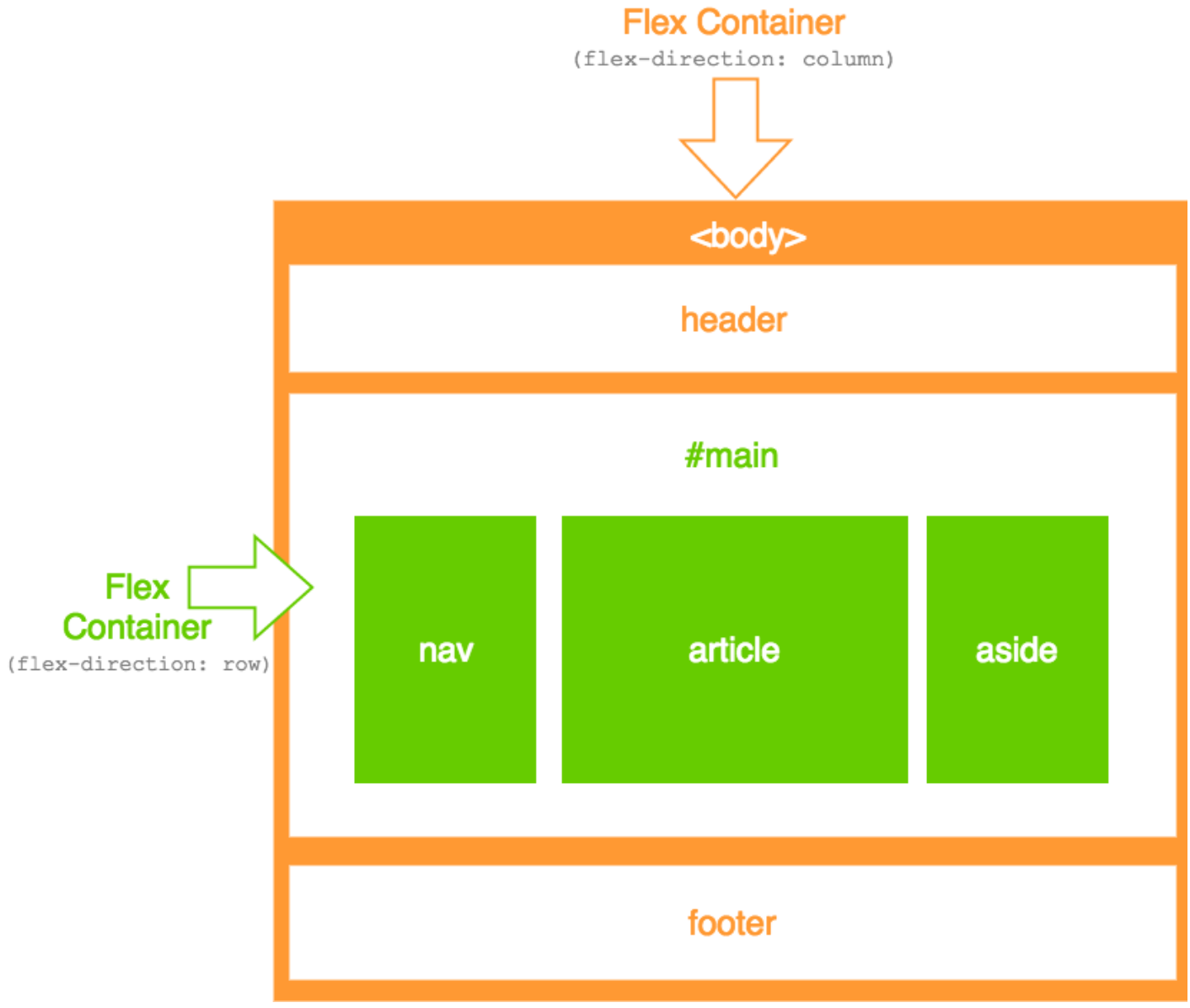


В данном примере мы применяем display: flex одновременно к внешнему контейнеру *и* к красному флекс-элементу. Но с красным флекс-элементом мы используем flex-direction: column, в результате чего его флекс-элементы располагаются вертикально друг под другом. Значением по умолчанию для flex-direction является row, поэтому мы не стали использовать это свойство для внешнего контейнера — по умолчанию флекс-элементы располагаются в ряд.

Двумерные макеты сайтов

Когда мы разбирали пример макета Святого Грааля, то использовали флексбоксы только для средней части, а шапку и подвал оставили блочными элементами. Мы применяли флексбоксы для размещения трёх флекс-элементов по горизонтали (в ряд), но не задавали флекс-элементы по вертикали (в колонку). Другими словами, у нас был одномерный макет. Несмотря на то, что страница в целом представляла собой двумерный макет, флексбоксы поддерживают макет только в одном из этих измерений.

Мы могли бы изменить тот пример для использования флексбоксов в двух измерениях — по рядам *и* колонкам. Для этого воспользуемся концепцией, показанной выше. Таким образом, мы можем использовать один флекс-контейнер для вертикальной компоновки, а другой — для горизонтальной.



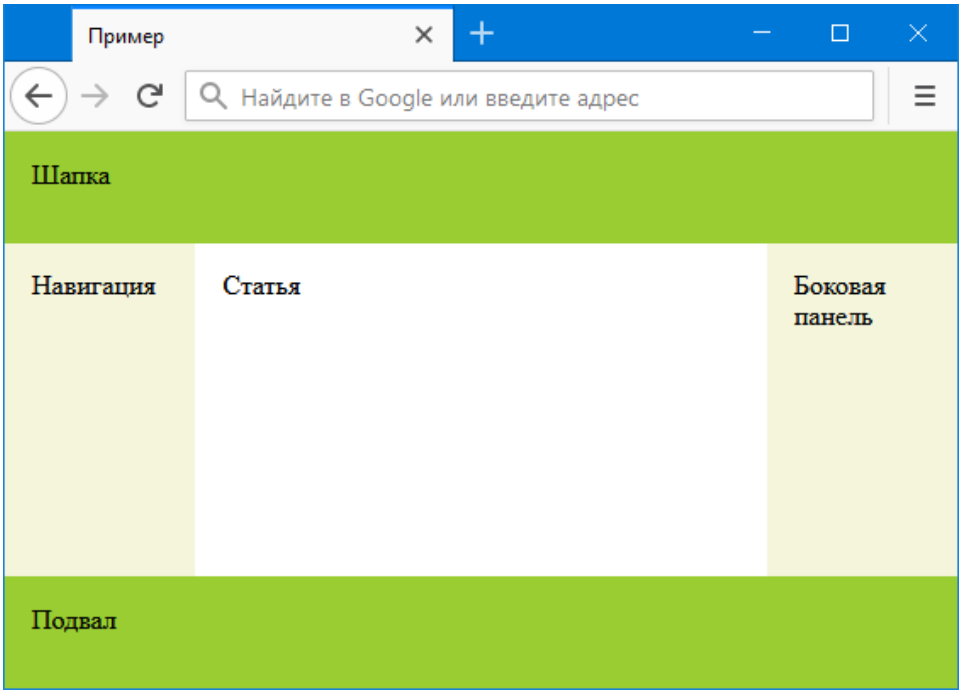
В нашем случае вертикальный макет состоит из следующих областей: шапка, основная часть, подвал. Вы можете представить всю страницу как одну большую колонку. И эта колонка содержит три ряда — один для шапки, один для основного содержимого и один для подвала.

Горизонтальный макет состоит из <nav>, <article> и <aside>. Все они содержатся во втором ряду, но их необходимо разместить горизонтально. Следовательно, мы можем сделать второй ряд флекс-контейнером и заставить его содержимое выстроиться в ряд.

Итак, вот рабочий пример.

☐ Тёмная тема

```
<!doctype html>
<title>Пример</title>
<style>
* {
  box-sizing: border-box;
}
body {
  display: flex;
  min-height: 100vh;
  flex-direction: column;
  margin: 0;
}
#main {
  display: flex;
  flex: 1;
}
#main > article {
  flex: 1;
}
#main > nav,
#main > aside {
  flex: 0 0 20vw;
  background: beige;
}
#main > nav {
  order: -1;
}
header, footer {
  background: yellowgreen;
  height: 20vh;
}
header, footer, article, nav, aside {
  padding: 1em;
}
</style>
<body>
  <header>Шапка</header>
  <div id="main">
    <article>Статья</article>
    <nav>Навигация</nav>
    <aside>Боковая панель</aside>
  </div>
  <footer>Подвал</footer>
</body>
```



В нашем случае мы использовали следующий код для элемента <body>.

```
body {
  display: flex;
  min-height: 100vh;
  flex-direction: column;
  margin: 0;
}
```

display: flex делает <body> флекс-контейнером, а flex-direction: column отображает его флекс-элементы в колонку. Мы также задали минимальную высоту для внешнего флекс-контейнера, чтобы макет занимал всю высоту экрана.

А теперь для вложенного флекс-контейнера.

```
#main {
  display: flex;
  flex: 1;
}
```

Добавление display: flex делает #main флекс-контейнером, подобно <body>. Теперь его дочерние элементы становятся флекс-элементами. flex: 1 гарантирует, что он будет расти так, чтобы занимать максимум доступного пространства.

Мы могли бы добавить flex-direction: row, чтобы явно указать направление, но в любом случае row является значением по умолчанию.

Остальное — просто стилизация и упорядочивание флекс-элементов во вложенном флекс-контейнере.

```
#main > article {
  flex: 1;
}
#main > nav,
#main > aside {
  flex: 0 0 20vw;
  background: beige;
}
#main > nav {
  order: -1;
}
```

Вы можете спросить, зачем использовать вложенные флекс-контейнеры, когда можно просто сделать одномерный макет, как в предыдущем примере. В большинстве случаев вам, вероятно, не придётся применять вложенные флекс-контейнеры, тем не менее этот метод может пригодиться при использовании адаптивного веб-дизайна. Когда весь макет сделан на флексбоксах, то это предоставляет больше возможностей при использовании медиа-запросов для отображения разных макетов на разных устройствах и т. д. Без этого вы можете использовать всю мощь флексбоксов только для части веб-сайта.

Кроме того, я уверен, вы можете придумать множество других применений для вложенных флекс-контейнеров, помимо макетов веб-сайтов.

Вложенные флексбоксы против CSS Grid

Каждый раз, когда используете вложенные флекс-контейнеры, подумайте, не лучше ли вместо этого использовать CSS Grid. Он специально разработан для двумерных макетов, поэтому для создания таких макетов не требуется вложенность (хотя она поддерживается).

Тем не менее, существует много макетов, для которых лучше подходят флексбоксы, так что рассмотрим несколько случаев.