Адаптивный макет на флексбоксах

Медиа-запросы обычно используются в адаптивном дизайне для отображения разных макетов на разных устройствах, в зависимости от размера экрана. Для этого есть причина — некоторые макеты всегда будут выглядеть слишком сплющенными (или полностью разломаются) при просмотре на узких экранах.

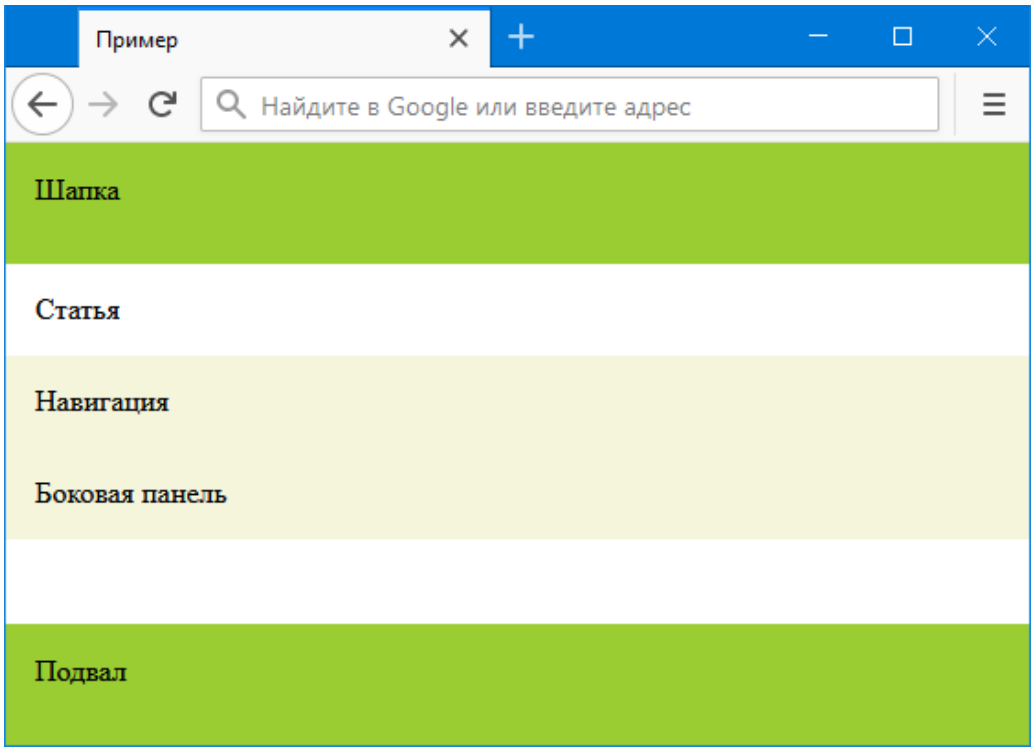
Например, вы можете использовать медиа-запросы для отображения первого макета на широких экранах (настольные компьютеры, ноутбуки и др), второго макета на экранах среднего размера (планшеты, большие телефоны) и третьего макета на узких экранах (мобильные телефоны и др.).

Здесь мы возьмём созданный нами ранее макет и добавим к нему медиа-запрос, чтобы он отображался по-другому на небольших устройствах.

☐ 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;
}
@media screen and (max-width: 575px) {
  #main {
    display: block;
  }
}
</style>
<body>
  <header>Шапка</header>
```

```
  <div id="main">
    <article>Статья</article>
    <nav>Навигация</nav>
    <aside>Боковая панель</aside>
  </div>
  <footer>Подвал</footer>
</body>
```



В этом примере все элементы должны располагаться друг под другом. Если нет, то вы, вероятно, смотрите на широком экране. В этом случае уменьшите размер окна браузера до тех пор, пока не увидите свёрнутый макет.

В любом случае вот что мы добавили в код.

```
@media screen and (max-width: 575px) {
  #main {
    display: block;
  }
}
```

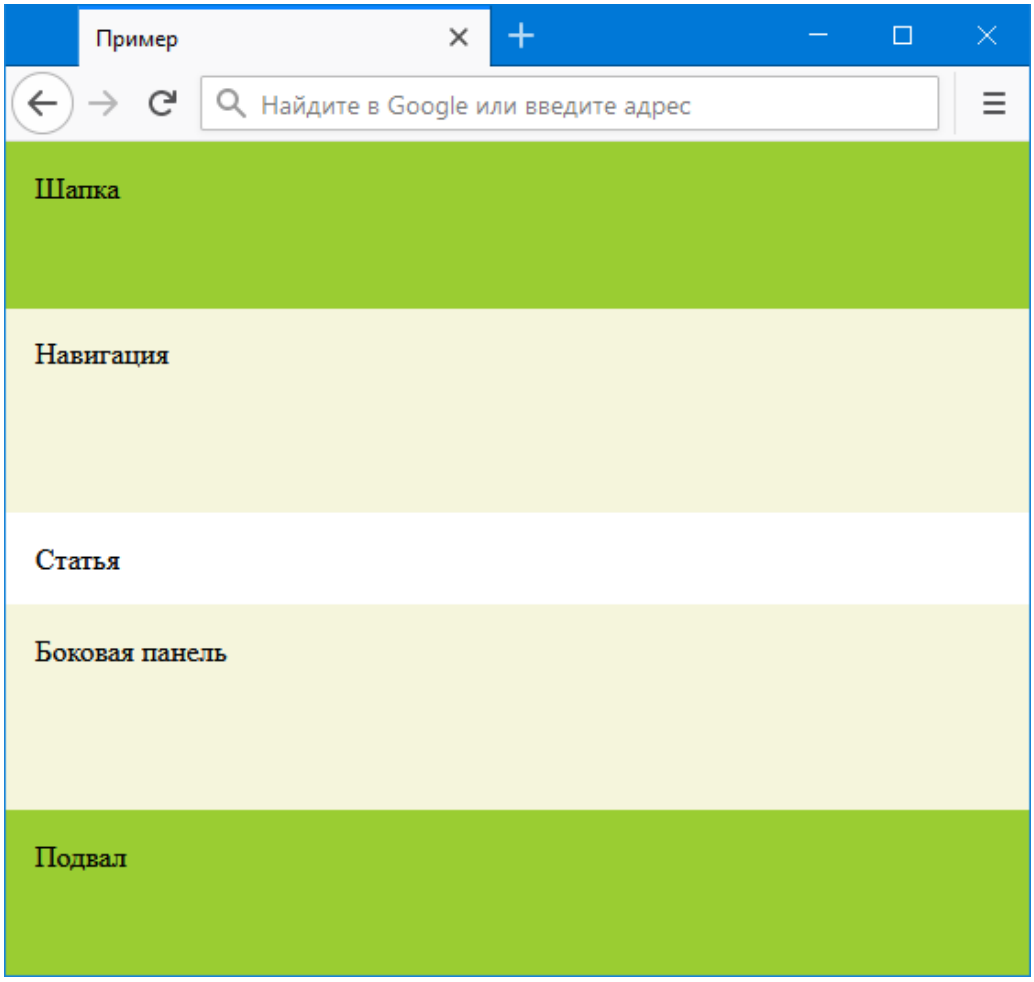
Всё, что мы здесь сделали, это изменили display: flex на display: block для элемента #main, чтобы его дочерние элементы перестали быть флекс-элементами. Это приводит к тому, что они накладываются друг на друга в исходном порядке.

Но что, если мы не желаем, чтобы элементы отображались в исходном порядке? Если мы хотим, чтобы навигационная панель появилась перед статьей? В этом случае мы можем внести столь же простые изменения.

☐ 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;
}
@media screen and (max-width: 575px) {
  #main {
    flex-direction: column;
  }
}
</style>
<body>
  <header>Шапка</header>
  <div id="main">
    <article>Статья</article>
    <nav>Навигация</nav>
    <aside>Боковая панель</aside>
  </div>
  <footer>Подвал</footer>
</body>
```



Вот что мы сделали взамен.

```
@media screen and (max-width: 575px) {
  #main {
    flex-direction: column;
```

```
  }
}
```

Итак, теперь у нас идёт навигационная панель, статья, затем боковая панель. Но заметьте, что навигация и боковая панель располагаются выше, чем в предыдущем примере. Это странно!

На самом деле, так происходит из-за этого фрагмента кода.

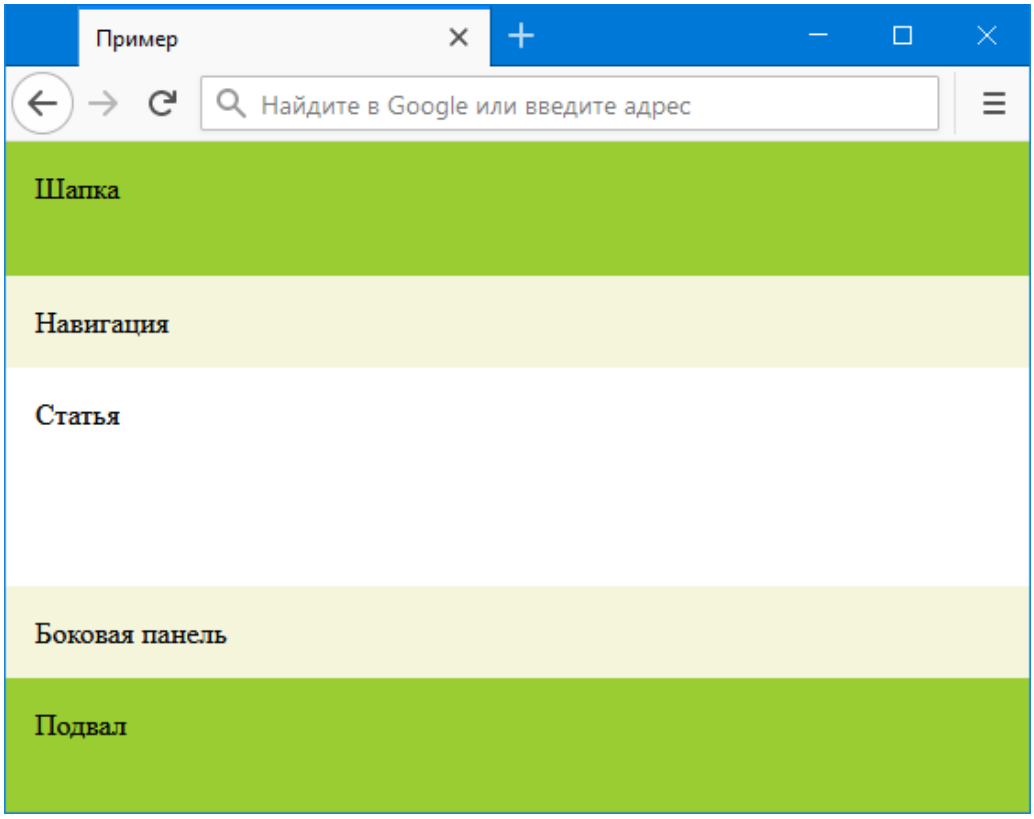
```
#main > nav,
  #main > aside {
  flex: 0 0 20vw;
  background: beige;
}
```

В частности, код flex: 0 0 20vw устанавливает для flex-base значение 20vw, что составляет 20% от ширины области просмотра. Ранее мы применяли это значение для ширины элементов, но теперь, когда значение flex-direction задано как column, оно используется для высоты.

Если мы хотим, чтобы высота была основана на содержимом, то можем изменить это значение на auto или вообще удалить строку.

☐ 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 {
  background: beige;
}
#main > nav {
  order: -1;
}
header, footer {
  background: yellowgreen;
  height: 20vh;
}
header, footer, article, nav, aside {
  padding: 1em;
}
@media screen and (max-width: 575px) {
  #main {
    flex-direction: column;
  }
}
</style>
<body>
  <header>Шапка</header>
  <div id="main">
    <article>Статья</article>
    <nav>Навигация</nav>
    <aside>Боковая панель</aside>
  </div>
  <footer>Подвал</footer>
</body>
```



Сначала мобильные

Мы могли бы переключить код так, чтобы наш макет стал «сначала мобильные». Это термин применяется для макетов, которые создаются в основном для мобильных устройств, но включают медиа-запрос, который меняет макет для более крупных устройств. Это противоположно тому, что мы делали выше, где наш макет по умолчанию был для больших устройств, а мы добавили медиа-запрос для маленьких устройств.

Таким образом, мы можем взять приведённый выше пример и изменить его следующим образом.

☐ 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;
  flex-direction: column;
}
#main > article {
  flex: 1;
}
#main > nav,
#main > aside {
  background: beige;
}
#main > nav {
  order: -1;
}
header, footer {
  background: yellowgreen;
  height: 20vh;
}
```

```
header, footer, article, nav, aside {
  padding: 1em;
}
@media screen and (min-width: 576px) {
  #main {
    flex-direction: row;
  }
  #main > nav,
  #main > aside {
    flex: 0 0 20vw;
  }
}
</style>
<body>
  <header>Шапка</header>
  <div id="main">
    <article>Статья</article>
    <nav>Навигация</nav>
    <aside>Боковая панель</aside>
  </div>
  <footer>Подвал</footer>
</body>
```

Макет по-прежнему выглядит так же (что хорошо), но наш медиа-запрос теперь другой.

```
@media screen and (min-width: 576px) {
  #main {
    flex-direction: row;
  }
  #main > nav,
  #main > aside {
    flex: 0 0 20vw;
  }
}
```

А весь остальной код идёт перед ним.

Обратите внимание, что медиа-запрос на этот раз использует min-width, чтобы соответствовать всем устройствам указанной ширины и больше. В предыдущем примере мы использовали max-width, чтобы соответствовать только устройствам, которые были указанной ширины или меньше.

Итак, мы установили для начального макета значение flex-direction как column, а для больших устройств как row. Мы также вернули обратно flex-basis для больших устройств.