

АДАПТИВНАЯ ВЕРСТКА



ДМИТРИЙ РЯБОВ



ДМИТРИЙ РЯБОВ

Front-end разработчик в Tactise

 vk.com/dimarikry

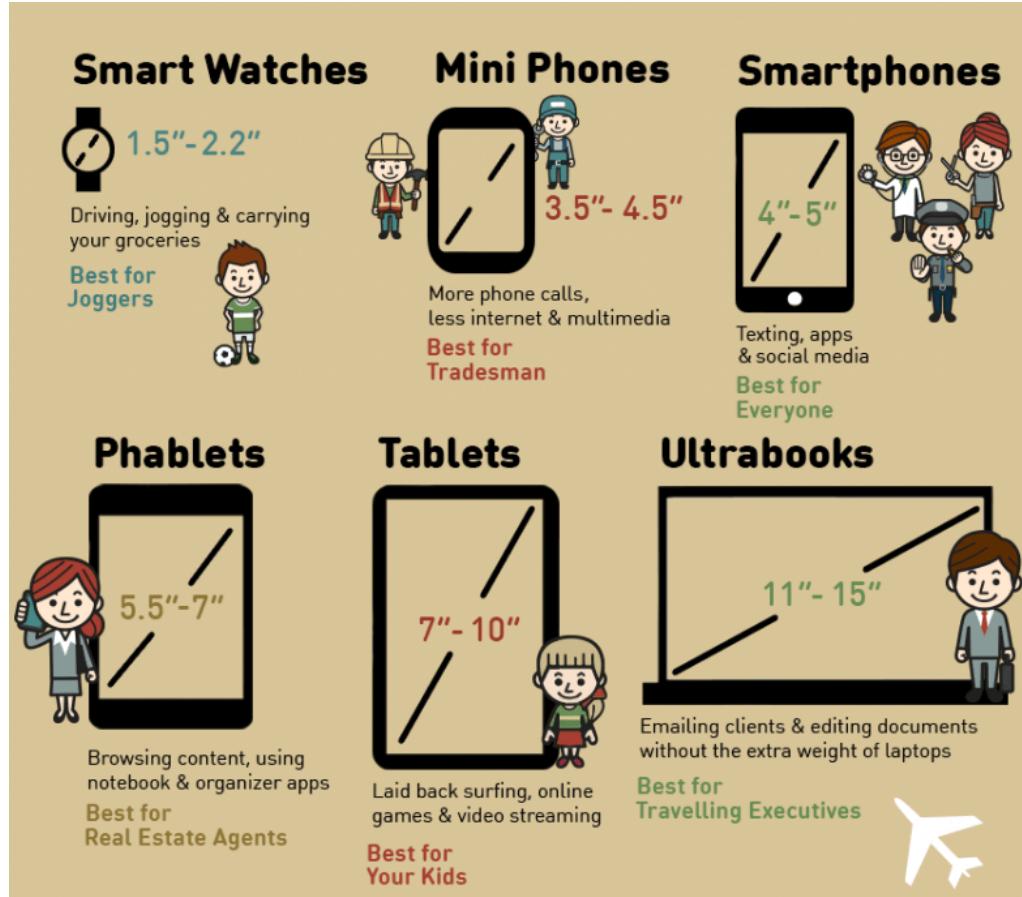
 fb.me/dmitriy.ryabov.7

ПЛАН ЗАНЯТИЯ

1. Адаптивная верстка
2. Логические операторы «И», «ИЛИ», «НЕ»
3. Адаптация фоновых изображений

АДАПТИВНАЯ ВЕРСТКА

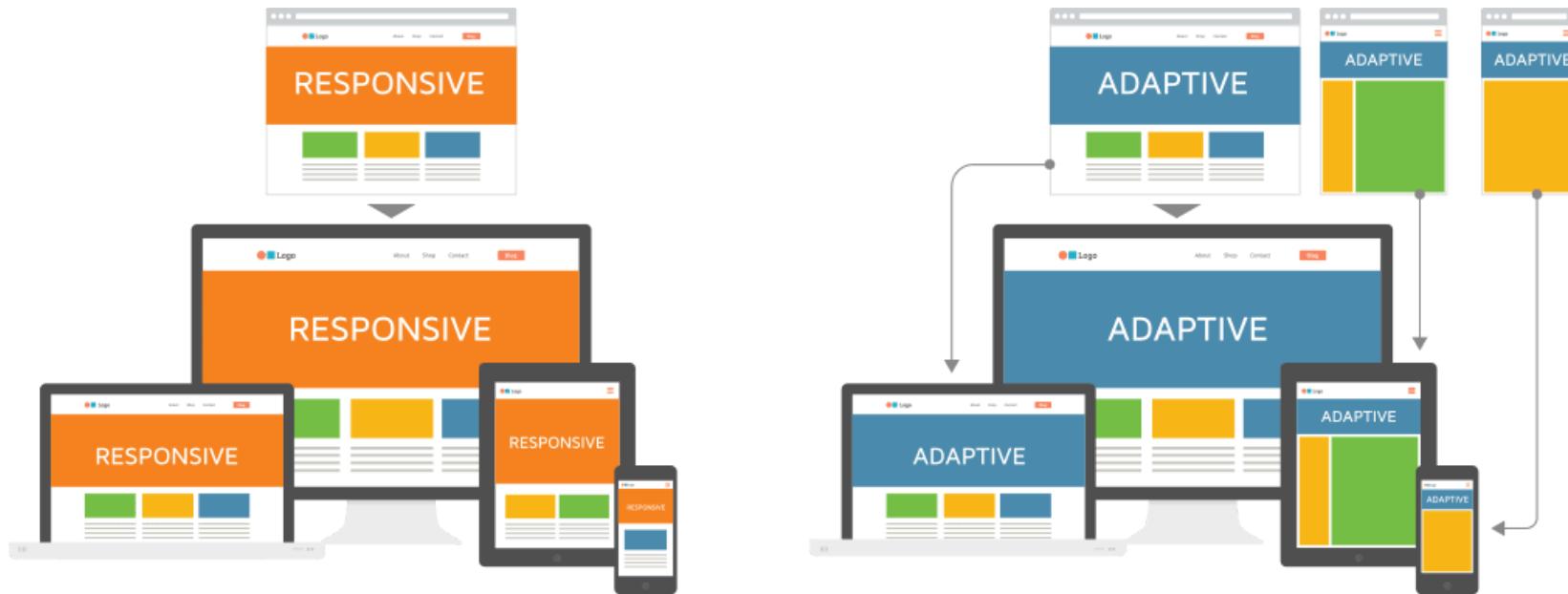
РАЗНООБРАЗИЕ УСТРОЙСТВ



И это далеко не все! [Полная инфографика](#)

УДОБНЫЙ ИНТЕРФЕЙС ВСЕГДА

Важно, чтобы страница не только красиво выглядела на каждом экране, но и была удобна в использовании. Эту задачу нельзя решить только на основе принципов резиновой верстки.

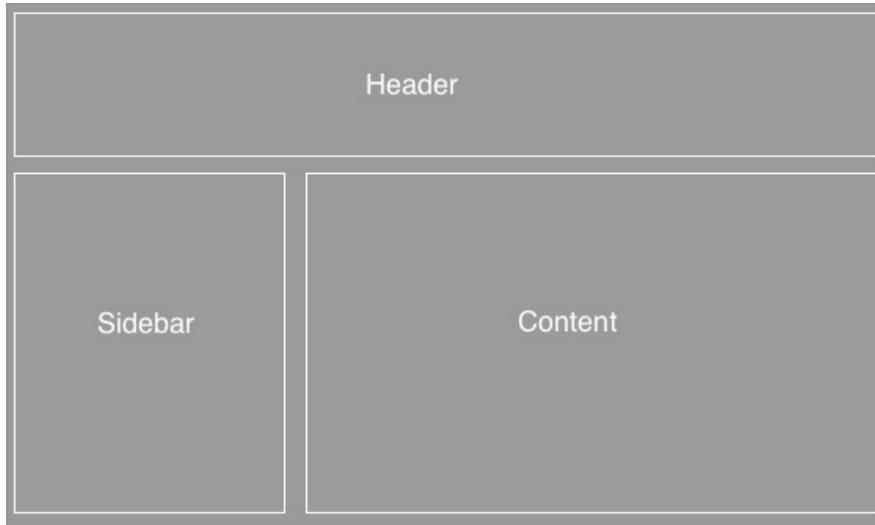




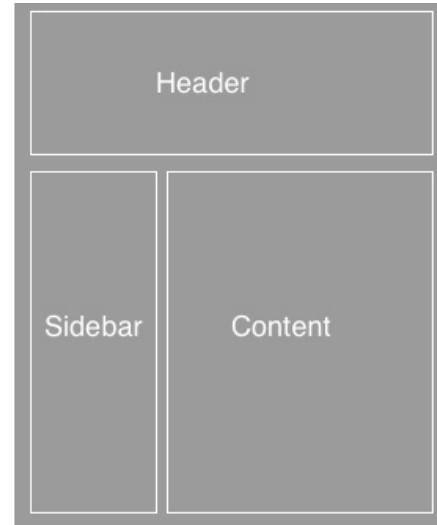
Адаптивная верстка – это такая верстка, при которой предполагается изменение дизайна в зависимости от размера экрана и ориентации устройства для того, чтобы максимально эффективно отобразить контент.



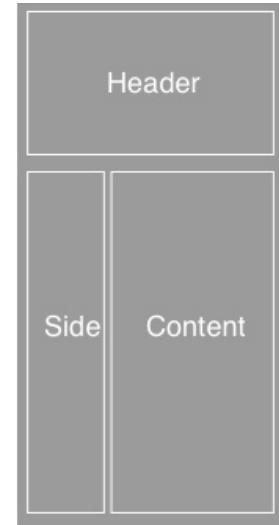
РЕЗИНОВАЯ ВЕРСТКА



1440px

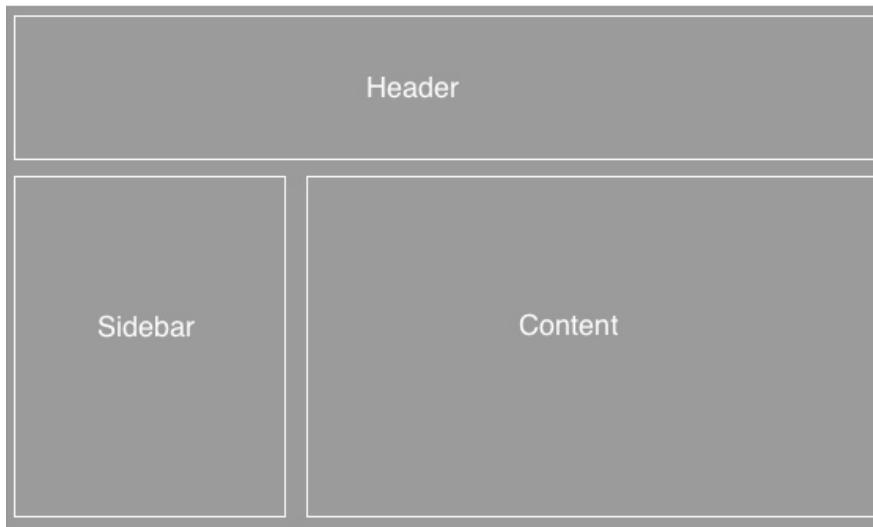


800px

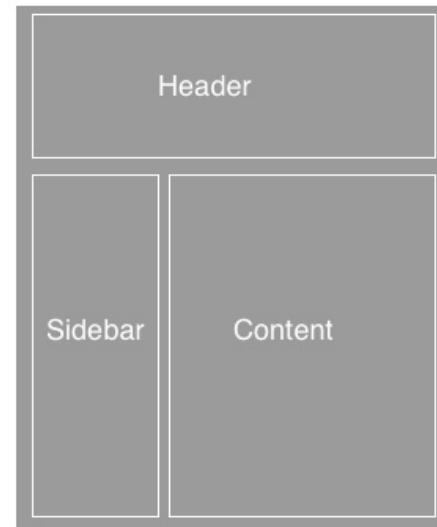


320px

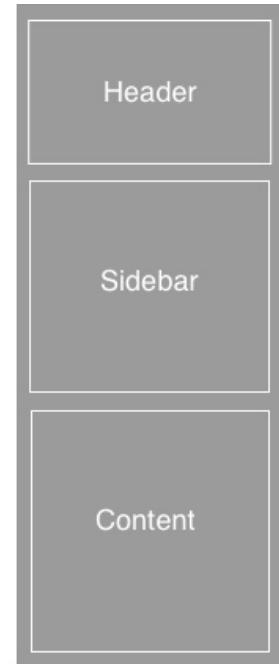
АДАПТИВНАЯ ВЕРСТКА



1440px



800px



320px

ПРИЕМЫ АДАПТИВНОГО ДИЗАЙНА

smashingmagazine.com на экране шириной 1280px:

The screenshot shows the homepage of smashingmagazine.com with a red header and a white content area. The header features a navigation bar with icons and text for Articles, Books, Events, Jobs, and Membership, along with a search bar for Topics. Below the header, a large white box contains the text "Don't Miss These Latest Articles". The main content area displays two articles: one by Rachel Andrew about getting started with CSS layout, and another by Vitaly Friedman about SmashingConf Toronto 2018. Each article has a thumbnail, the author's name, and a "Comments" section. A small "Seriously, red?" button is visible on the right side.

Articles
Design & development

Books
Physical & digital books

Events
Conferences & workshops

Jobs
Find work & employees

Membership
Webinars & early-birds

Topics

Don't Miss These Latest Articles

Rachel Andrew wrote

Vitaly Friedman wrote

Getting Started With CSS Layout 11 comments

CSS 250 # Layouts 42

A Conference Without Slides: Meet SmashingConf Toronto 2018 (June 26-27)

Seriously, red?

ШИРОКОФОРМАТНЫЙ МОНИТОР

Логотип меняется на полный и выводятся 3 превью последних записей:

The screenshot shows a desktop browser window displaying the Smashing Magazine website. The site has a red header with navigation links for Articles, Books, Events, Jobs, and Membership. A search bar labeled 'Topics' is also present. Below the header, there's a section titled 'Don't Miss These Latest Articles' featuring three recent posts by Rachel Andrew, Vitaly Friedman, and Ire Aderinokun. Each post includes a thumbnail, the author's name, the article title, a 'Leave a comment' button, and a list of tags at the bottom. The overall layout is designed to be viewed on a wide monitor.

SMASHING MAGAZINE

Articles
Design & development

Books
Physical & digital books

Events
Conferences & workshops

Jobs
Find work & employees

Membership
Webinars & early-birds

Topics

Don't Miss These Latest Articles

Rachel Andrew wrote
Getting Started With CSS Layout
11 comments
CSS 250 # Layouts 42

Vitaly Friedman wrote
A Conference Without Slides: Meet SmashingConf Toronto 2018 (June 26-27)
Leave a comment
Events 53 # SmashingConf 8

Ire Aderinokun wrote
Creating The Feature Queries Manager DevTools Extension
Leave a comment
Tools 173 # CSS 250 # JavaScript 190

ПЛАНШЕТ: ГОРИЗОНТАЛЬНАЯ ОРИЕНТАЦИЯ

Articles Books Events Jobs Membership Topics

Don't Miss These Latest Articles

Rachel Andrew wrote

Getting Started With CSS Layout 11 comments

CSS 250 # Layouts 42

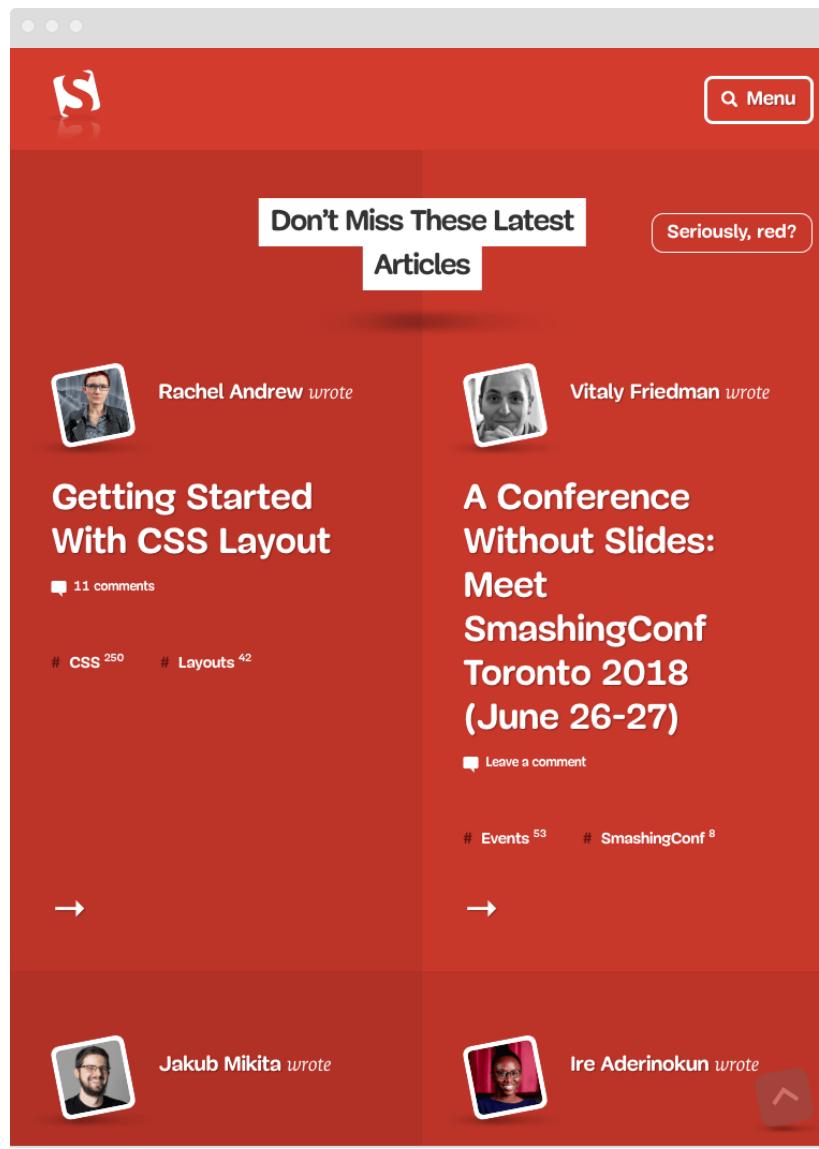
Vitaly Friedman wrote

A Conference Without Slides: Meet SmashingConf Toronto 2018 (June 26-27)

Leave a comment

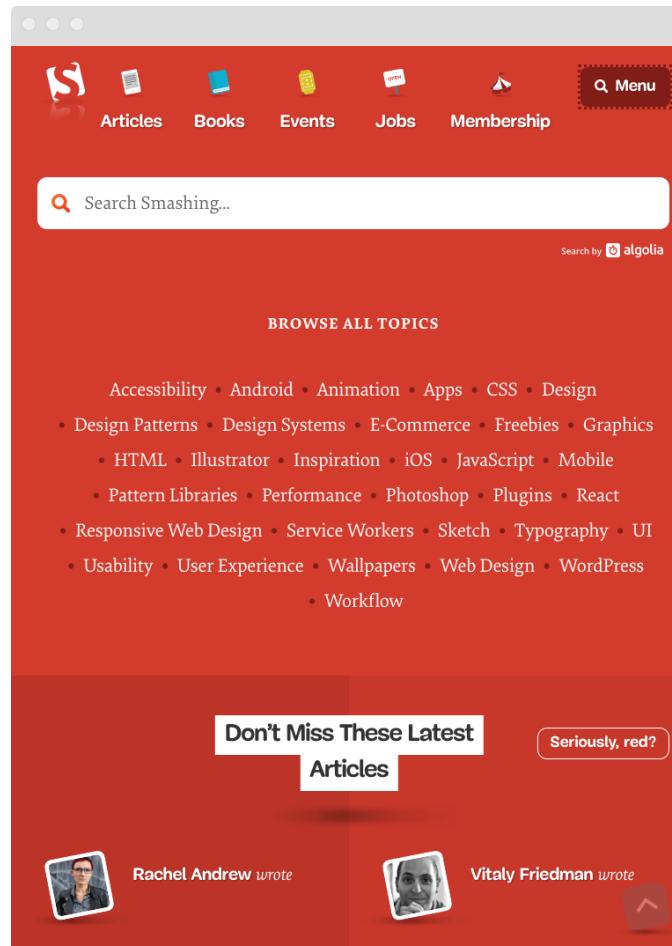
Events 53 # SmashingConf 8

ПЛАНШЕТ: ВЕРТИКАЛЬНАЯ ОРИЕНТАЦИЯ



ПЛАНШЕТ: ОТКРЫТОЕ МЕНЮ

В вертикальной ориентации поиск и меню прячутся за кнопками.
Категории «перемещаются» в выпадающее меню.



СМАРТФОН: ГОРИЗОНТАЛЬНО

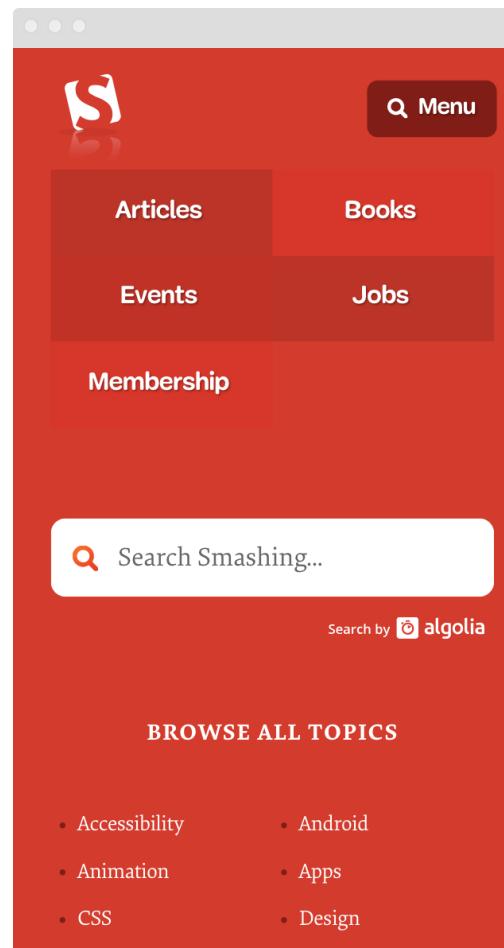
Основная часть состоит из одной колонки.



The image shows a smartphone screen displaying a website with a red background. At the top left is a white logo consisting of a stylized 'S' with a reflection below it. At the top right is a white button with a magnifying glass icon and the word 'Menu'. In the center of the screen, there is a white rectangular callout box containing the text 'Don't Miss These Latest Articles'. At the bottom left, there is a circular profile picture of a woman with glasses and dark hair, identified as 'Rachel Andrew wrote'. The overall layout is a single column design.

СМАРТФОН: ОТКРЫТОЕ МЕНЮ

Меню в вертикальной ориентации телефона состоит из одной колонки, а не из двух.



ПРИ ПРИЕМКЕ МАКЕТА

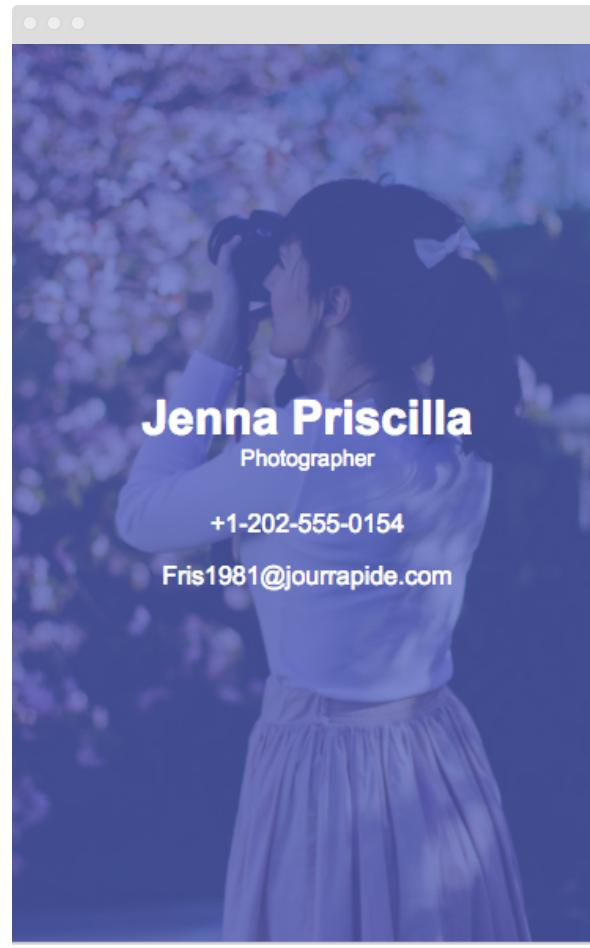
Важно внимательно изучить макеты и убедиться в том, что вам абсолютно точно понятно:

- Как перестраиваются элементы при сужении и расширении экрана.
- Какие элементы должны быть скрыты на маленьких экранах, и начиная с какой ширины, это должно происходить.
- Содержат ли скрывающиеся блоки изображения: если содержат, нужно помнить о том, что такие изображения все равно загружаются.

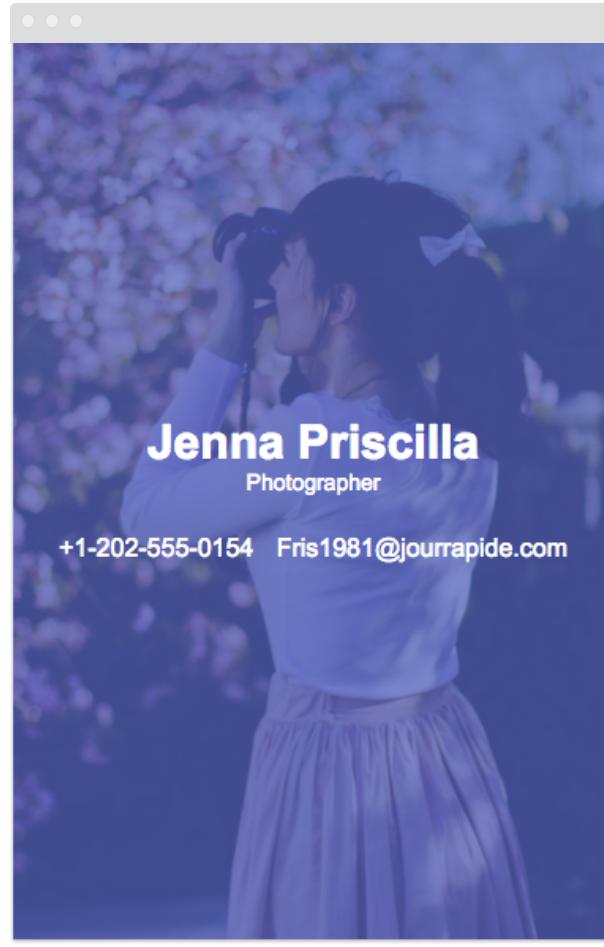
ЛОГИЧЕСКИЕ ОПЕРАТОРЫ «И», «ИЛИ», «НЕ»

КАРТОЧКА ПОЛЬЗОВАТЕЛЯ

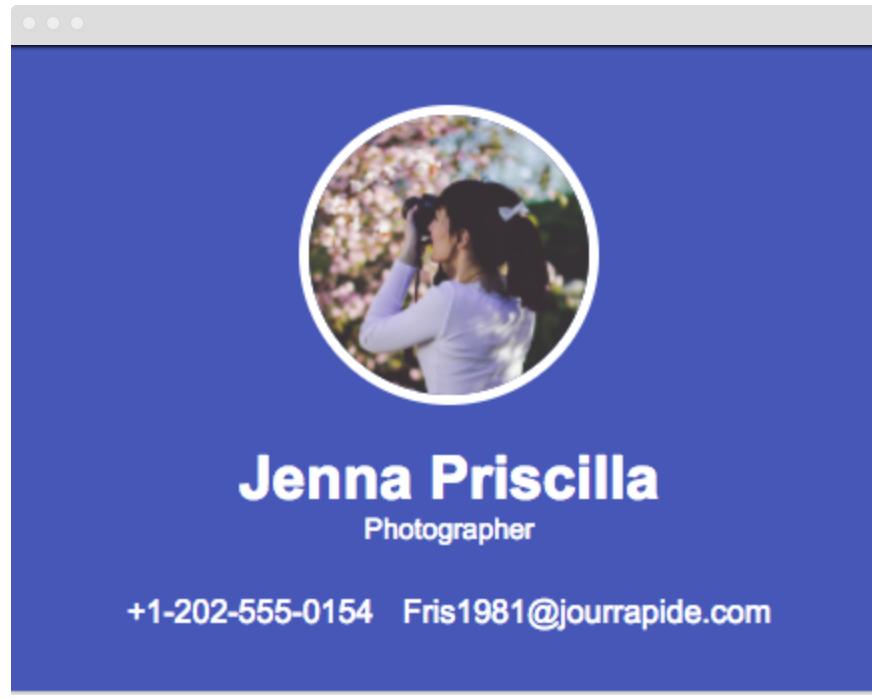
На последнем проекте нам поступила задача сверстать карточку пользователя с его контактами. Макет для мобильных:



МАКЕТ ДЛЯ ДЕСКТОПА



МАКЕТ ДЛЯ ПЛАНШЕТА



МАСКА НА ФОТО

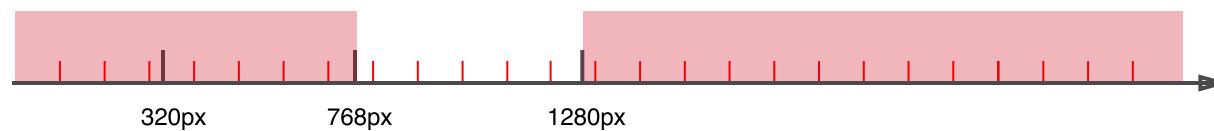
На фотографию накладывается полупрозрачная синяя маска, чтобы текст хорошо читался, даже если он попадет на светлые участки фото.

```
1 @media (max-width: 640px) {  
2     .profile__preview::before {  
3         content: "";  
4         width: 100%;  
5         height: 100%;  
6         background-color: #4557bb;  
7         opacity: 0.7;  
8     }  
9 }  
10 @media (min-width: 1301px) {  
11     .profile__preview::before {  
12         content: "";  
13         width: 100%;  
14         height: 100%;  
15         background-color: #4557bb;  
16         opacity: 0.7;  
17     }  
18 }
```

ЛОГИЧЕСКОЕ «ИЛИ» И ОПЕРАТОР ,

Смысл логического оператора запятая (,) соответствует логическому «ИЛИ». Он позволяет задать в медиазапросе несколько условий, истинности одного из которых будет достаточно для того, чтобы браузер применил CSS.

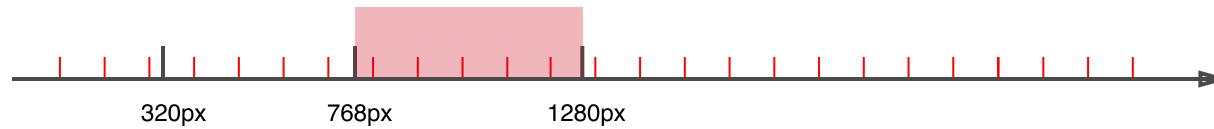
```
1 @media (max-width: 768px), (min-width: 1280px) {  
2   p {  
3     font-size: 1rem;  
4   }  
5 }
```



ЛОГИЧЕСКОЕ «И» И ОПЕРАТОР `and`

Оператор `and` позволяет указать несколько условий в медиазапросе, которые должны выполняться одновременно для того, чтобы стили были применены браузером:

```
1 @media (min-width: 768px) and (max-width: 1280px) {  
2   p {  
3     font-size: 1rem;  
4   }  
5 }
```



ЛОГИЧЕСКОЕ «НЕ» И ОПЕРАТОР `not`

С помощью логического `not` можно заставить медиазапрос срабатывать в случае, противоположном указанному условию:

```
1 @media not screen {  
2   .section {  
3     background: #cccccc;  
4   }  
5 }
```

Стили сработают только в тех случаях, когда устройство пользователя не является экраном.

В практике `not` используется достаточно редко, но когда встречается, то чаще всего с медиатипами.

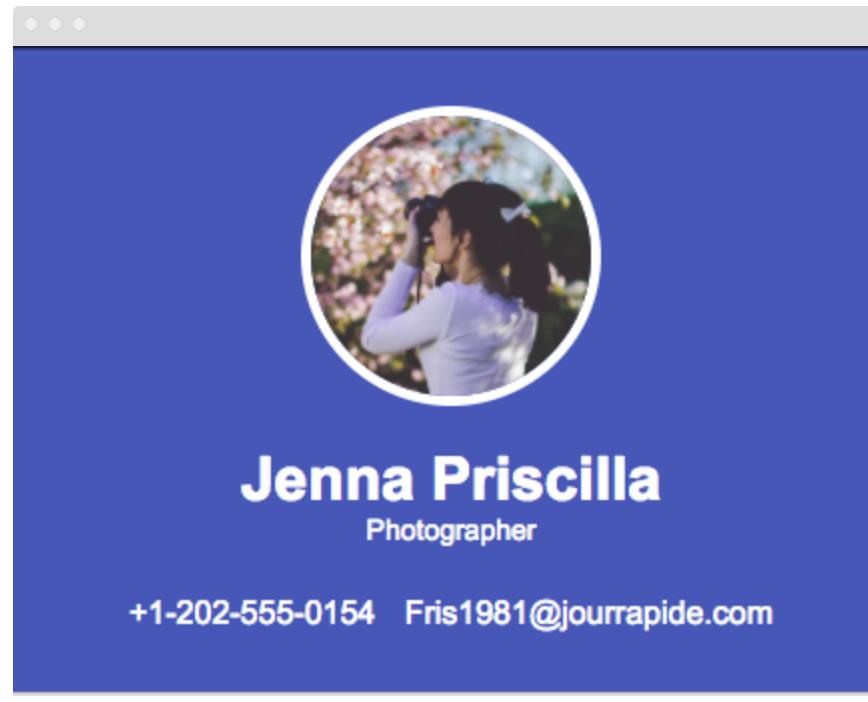
ОПТИМИЗИРУЕМ КОД МАСКИ

Перепишем код с использованием оператора `,`, чтобы избежать повторения:

```
1 @media (max-width: 640px), (min-width: 1301px) {  
2     .profile_preview::before {  
3         content: "";  
4         width: 100%;  
5         height: 100%;  
6         background-color: #4557bb;  
7         opacity: 0.7;  
8     }  
9 }
```

ДРУГОЙ ЮЗЕРПИК

В версии для планшетов у нас изменилось отображение фотографии пользователя:



Такое отображение требуется только для устройств с шириной экрана в диапазоне от 641px до 1300px.

МЕНЯЕМ ОТОБРАЖЕНИЕ ДЛЯ ПЛАНШЕТОВ

Для реализации этого условия как раз подходит логический оператор `and`:

```
1 @media (min-width: 641px) and (max-width: 1300px) {  
2     .profile_preview {  
3         width: 140px;  
4         height: 140px;  
5         border: 5px solid #ffffff;  
6         border-radius: 50%;  
7     }  
8 }
```

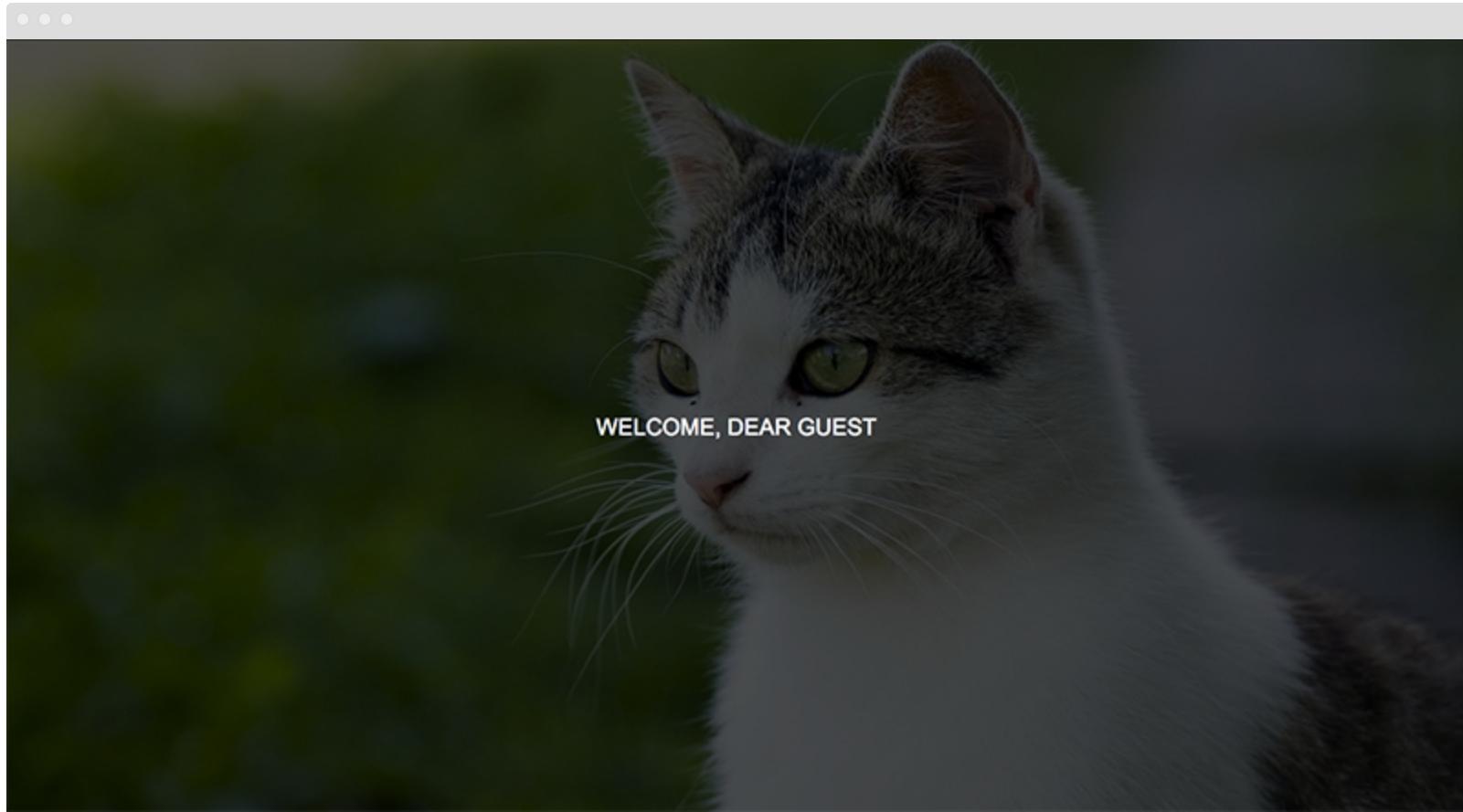
Теперь наша верстка выглядит точно так же, как в макете.

[Live Demo](#)

АДАПТАЦИЯ ФОНОВЫХ ИЗОБРАЖЕНИЙ

ЭКРАН ПРИВЕТСТВИЯ

Поступила задача сверстать экран с приветствием, которое будет занимать всю ширину и высоту окна браузера:



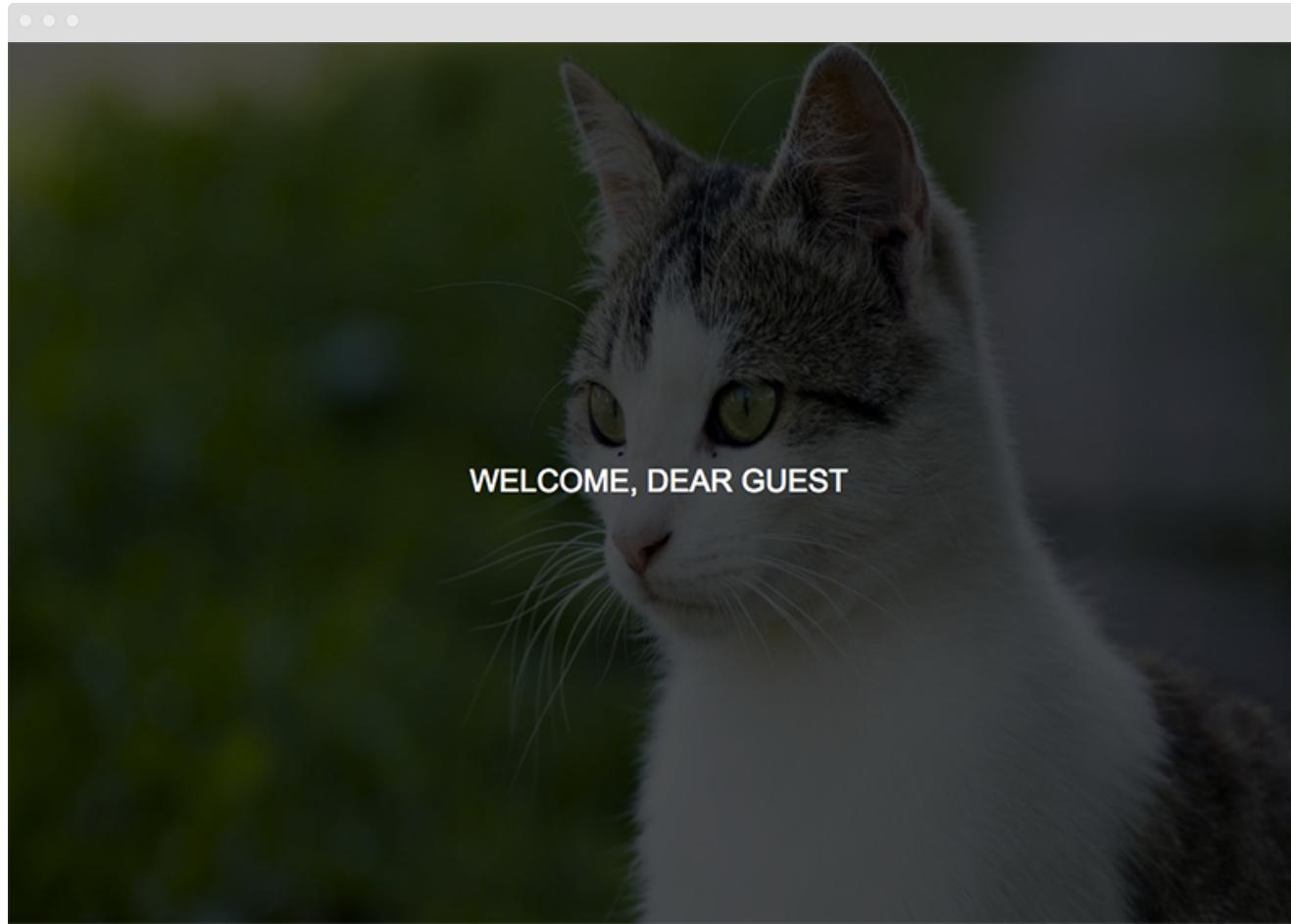
СТИЛИ ДЛЯ ЭКРАНА

Для того, чтобы изображение располагалось на всю ширину и высоту, мы указали свойство `background-size` в процентах:

```
1 .hero {  
2   background: url("cat.jpg") no-repeat 50% 50% #000000;  
3   background-size: 100% 100%;  
4 }
```

МОНИТОР ПОМЕНЬШЕ

На мониторе поменьше (1280*1024) картинки исказились:



ПРИЧИНА ДЕФОРМАЦИИ

Так как мы указали `background-size: 100% 100%`, то фоновая картинка тянется по ширине и высоте блока `.hero`, который, в свою очередь, принимает размеры экрана.

А вот пропорции экрана у нас изменились – экран 1920px*1080px и экран 1280px*1024px имеют разное соотношение высоты к ширине, поэтому наша картинка исказилась.

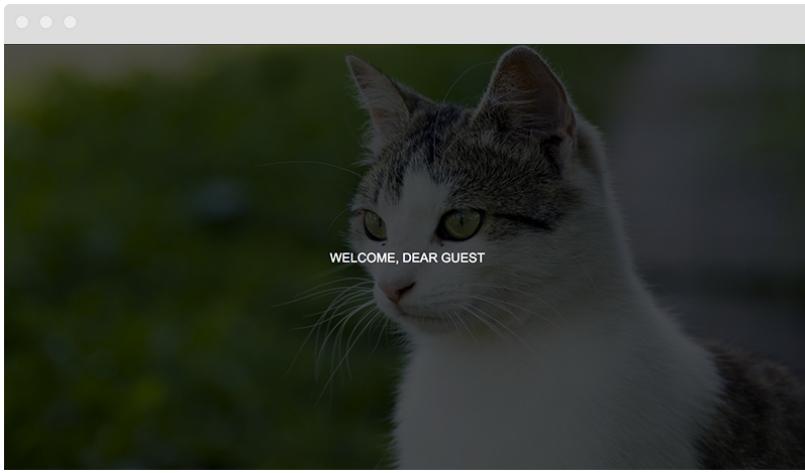
background-size: contain

При этом значении фоновое изображение всегда будет выводиться с сохранением пропорций, вписываясь в блок.

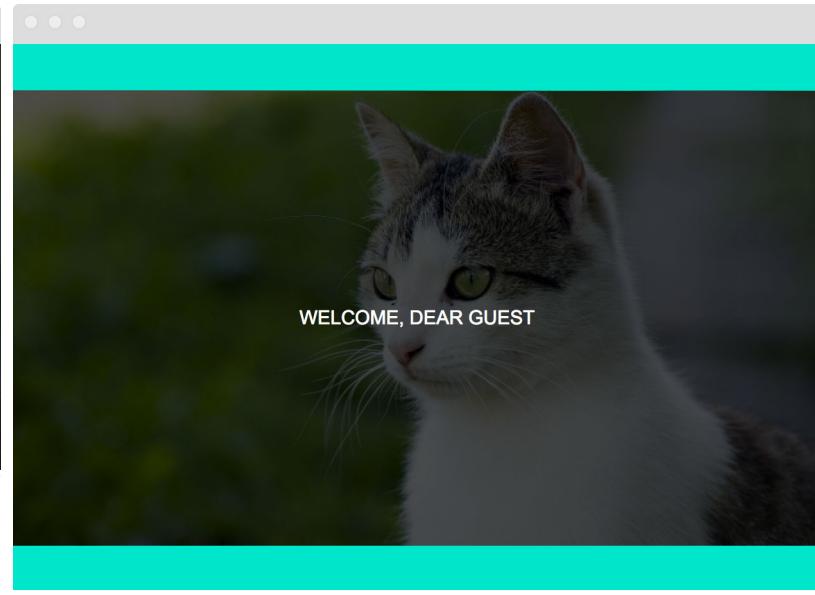
```
1 .hero {  
2   background: url("cat.jpg") no-repeat 50% 50% #000000;  
3   background-size: contain;  
4 }
```

РАЗНИЦА НА ЭКРАНАХ

1920px*1080px



1280px*1024px



Фоновое изображение не искажено. Но так как соотношение высоты и ширины картинки и блока заметно различаются, то сверху и снизу у нас возникли пустые полосы.

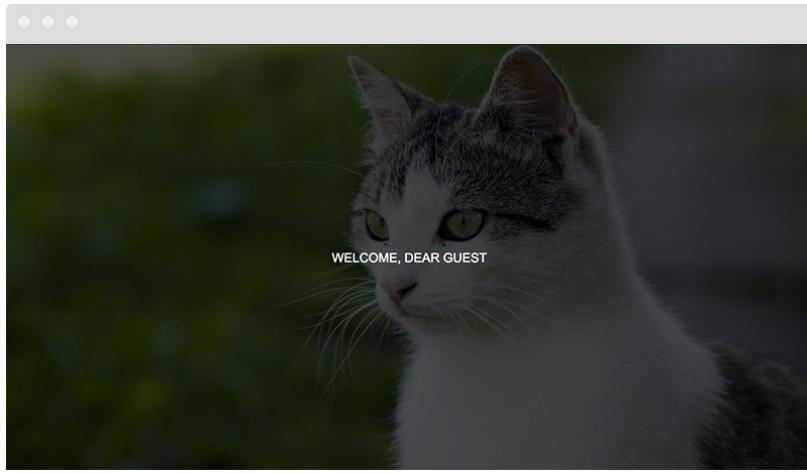
background-size: cover

При этом значении фоновое изображение будет выводиться с сохранением пропорций, но уже будет растягиваться или сжиматься таким образом, чтобы заполнить собой всю площадь блока.

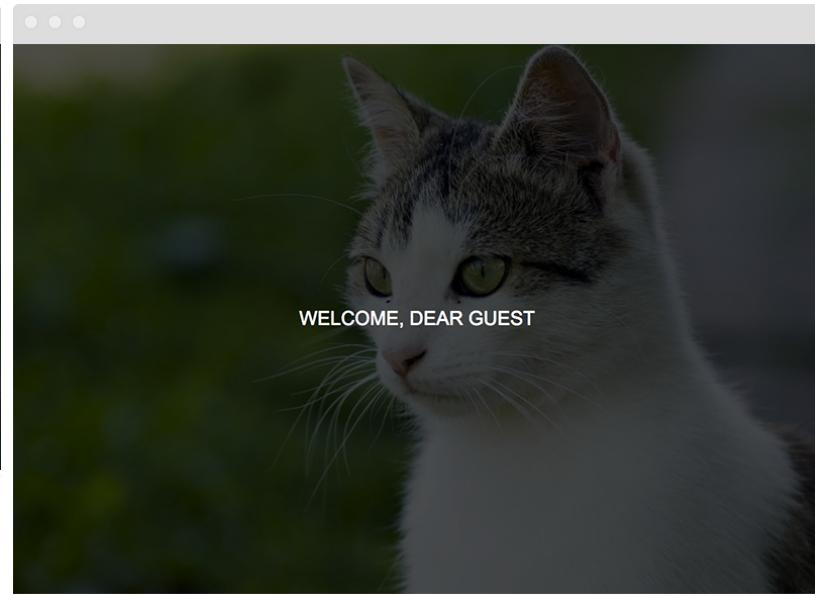
```
1 .hero {  
2   background: url("cat.jpg") no-repeat 50% 50% #000000;  
3   background-size: cover;  
4 }
```

ПРОВЕРЯЕМ НА РАЗНЫХ МОНИТОРАХ

1920px*1080px



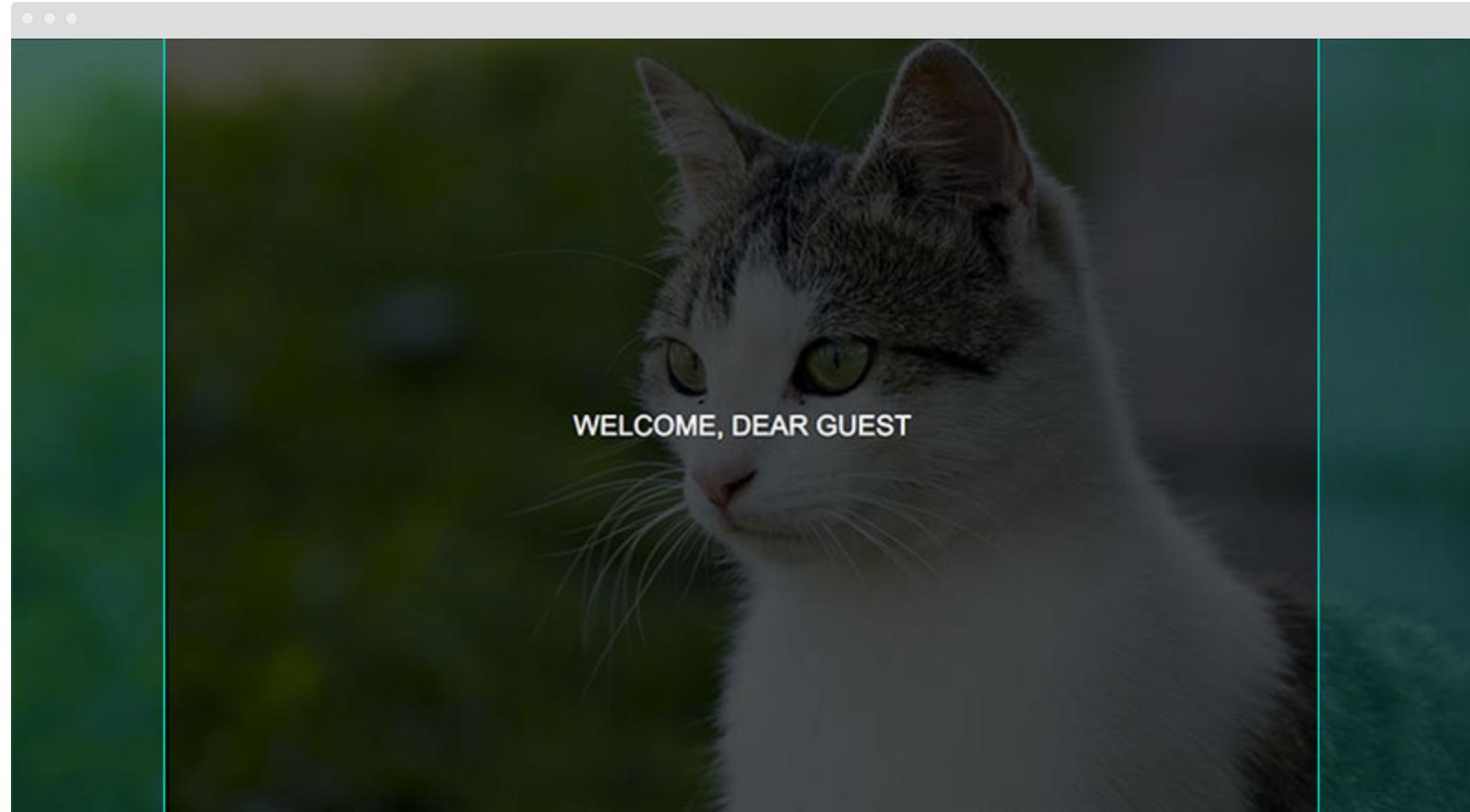
1280px*1024px



ОБРЕЗАЕМ ЛИШНЕЕ

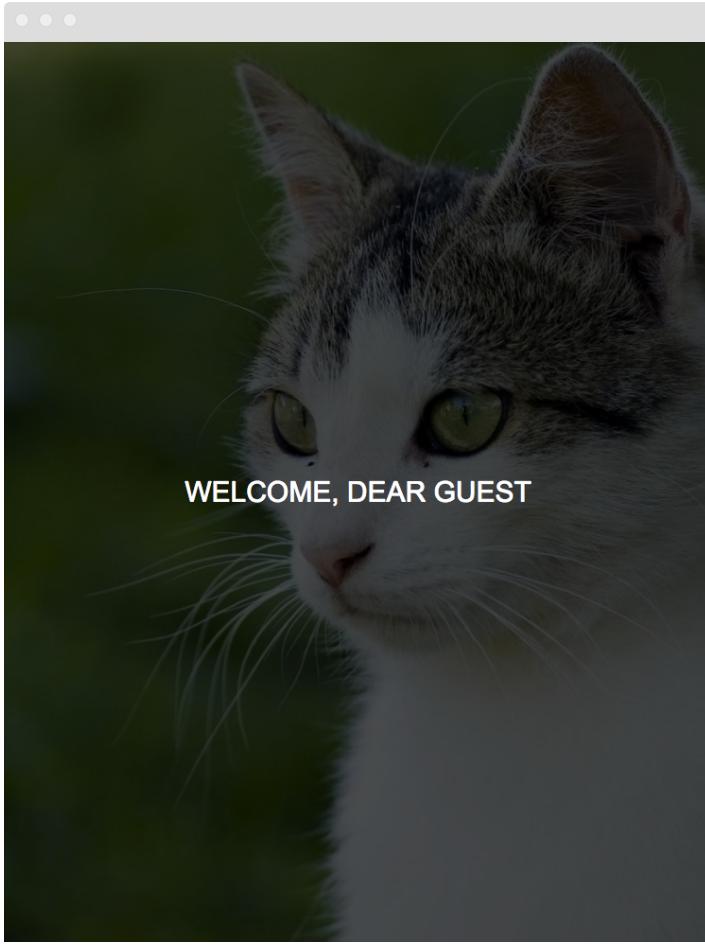
Картишка не искажена и при этом занимает наш блок целиком.

Изображение растянулось по высоте так, что «лишние» фрагменты остались невидимыми, оказались за пределами блока.

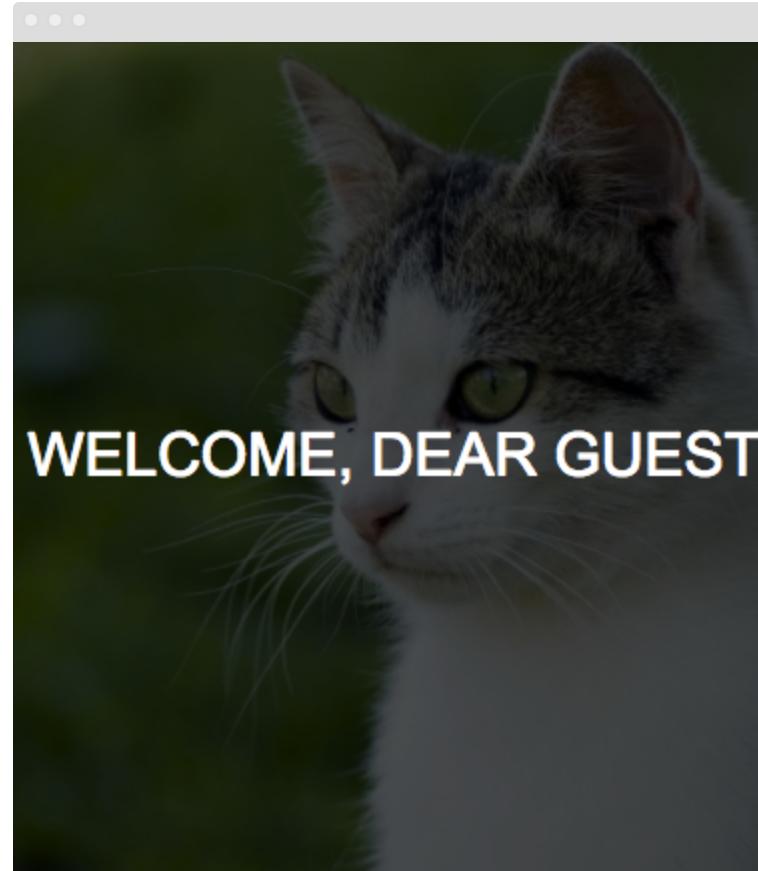


ОТЛИЧНЫЙ ФОН НА ЛЮБОМ УСТРОЙСТВЕ

1024px*768px



320px*480px



ПОМНИМ О РАЗМЕРЕ

Наше первоначальное изображение было подготовлено для размера экрана 1920px*1080px и «весит» 159 Кб. Помимо того, что при нестабильном соединении картинка такого размера может грузиться несколько секунд, на экране мобильного телефона большое изображение попросту не нужно, ведь на маленьком экране разница незаметна.

РАЗНЫЕ КАРТИНКИ ПОД ВСЕ РАЗРЕШЕНИЯ

Чаще всего дизайнер подготавливает несколько вариантов фонового изображения и отдает их вместе с макетом, а нам лишь остается подключить их, используя медиазапросы:

```
1 .hero {  
2     background-repeat: no-repeat;  
3     background-position: 50% 50%;  
4     background-color: #000000;  
5     background-size: cover;  
6 }  
7  
8 @media (max-width: 480px) {  
9     .hero {  
10         background-image: url("cat_480.jpg");  
11     }  
12 }  
13  
14 @media (min-width: 481px) and (max-width: 767px) {  
15     .hero {  
16         background-image: url("cat_767.jpg");  
17     }  
18 }
```

ОСТАЛЬНЫЕ РАЗРЕШЕНИЯ

```
1 @media (min-width: 768px) and (max-width: 1279px) {  
2     .hero {  
3         background-image: url("cat_1279.jpg");  
4     }  
5 }  
6  
7 @media (min-width: 1280px) and (max-width: 1439px) {  
8     .hero {  
9         background-image: url("cat_1439.jpg");  
10    }  
11 }  
12  
13 @media (min-width: 1440px){  
14     .hero {  
15         background-image: url("cat.jpg");  
16     }  
17 }
```

ИТОГИ

АДАПТИВНАЯ ВЕРСТКА

- Страница не только должна красиво выглядеть на каждом экране, но и быть удобной в использовании. Эту задачу нельзя решить только на основе принципов резиновой верстки.
- Адаптивная верстка – это верстка, при которой предполагается изменение дизайна в зависимости от размера экрана и ориентации устройства для того, чтобы максимально эффективно отобразить контент.
- Внимательно изучаем макеты, выясняем, как перестраиваются элементы при сужении и расширении экрана, какие элементы должны быть скрыты на маленьких экранах, и, начиная с какой ширины, это должно происходить.
- Проверяем, подготовлены ли все нужные для адаптивной верстки изображения. Если нет – запрашиваем их.

ЛОГИЧЕСКИЕ ОПЕРАТОРЫ «И», «ИЛИ», «НЕ»

- Логический оператор запятая (,) соответствует логическому «ИЛИ». Он позволяет задать в медиазапросе несколько условий, истинности одного из которых будет достаточно для того, чтобы браузер применил CSS.
 - Оператор `and` позволяет указать несколько условий в медиазапросе, которые должны выполняться одновременно для того, чтобы стили были применены браузером.
 - С помощью логического `not` можно заставить медиазапрос срабатывать в случае, противоположном указанному условию (в практике используется достаточно редко).
-

АДАПТАЦИЯ ФОНОВЫХ ИЗОБРАЖЕНИЙ

- `background-size: contain;` – фоновое изображение выводится с сохранением пропорций, вписываясь в блок.
- `background-size: cover;` – фоновое изображение выводится с сохранением пропорций, при этом растягивается/сжимается таким образом, чтобы заполнить собой всю площадь блока.
- Чаще всего дизайнер подготавливает несколько вариантов фонового изображения и отдает их вместе с макетом. Мы подключаем их, используя медиазапросы.



Задавайте вопросы и напишите отзыв о лекции!

ДМИТРИЙ РЯБОВ

 vk.com/dimarikrv

 fb.me/dmitriy.ryabov.7