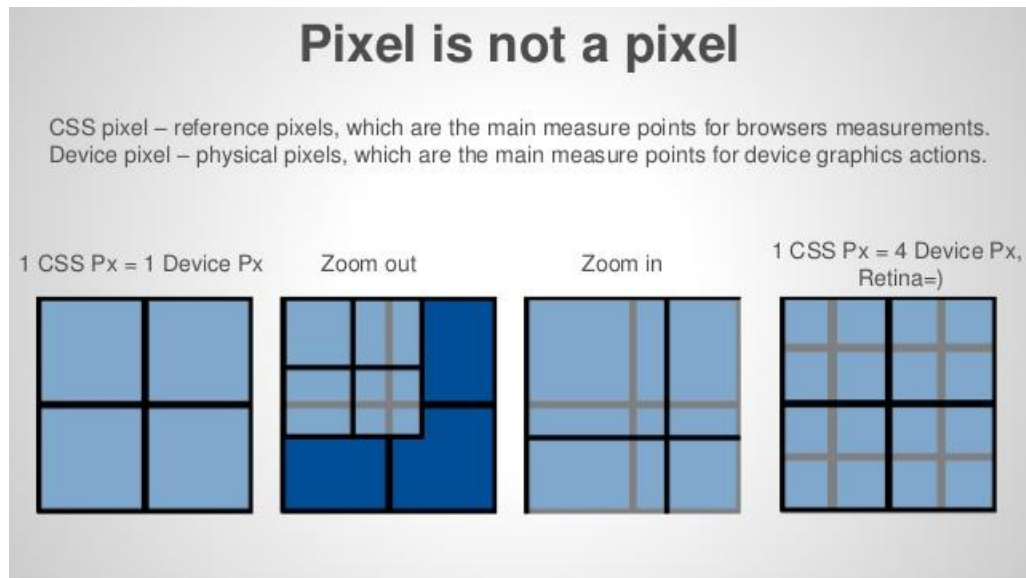


# Адаптивная графика. Retina.



# Экранные и CSS-пиксели

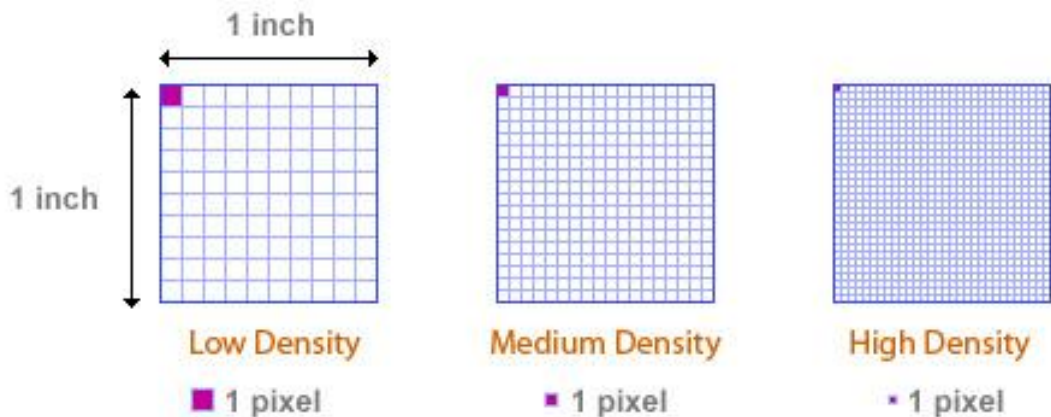
Физические и CSS-пиксели — разные понятия, их размеры редко совпадают. Это может происходить при зуме или на современных экранах (ретина).



# Retina

Экраны с повышенной плотностью пикселей.

Количество физических пикселей больше, чем у обычного экрана, а сами пиксели меньше.



**CSS**  
pixels

Standard

height: 2px  
width: 2px

==

Retina

height: 2px  
width: 2px

**Bitmap**  
pixels



x4

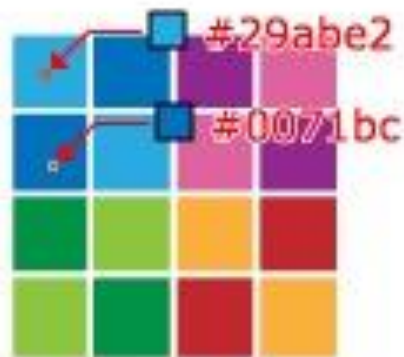


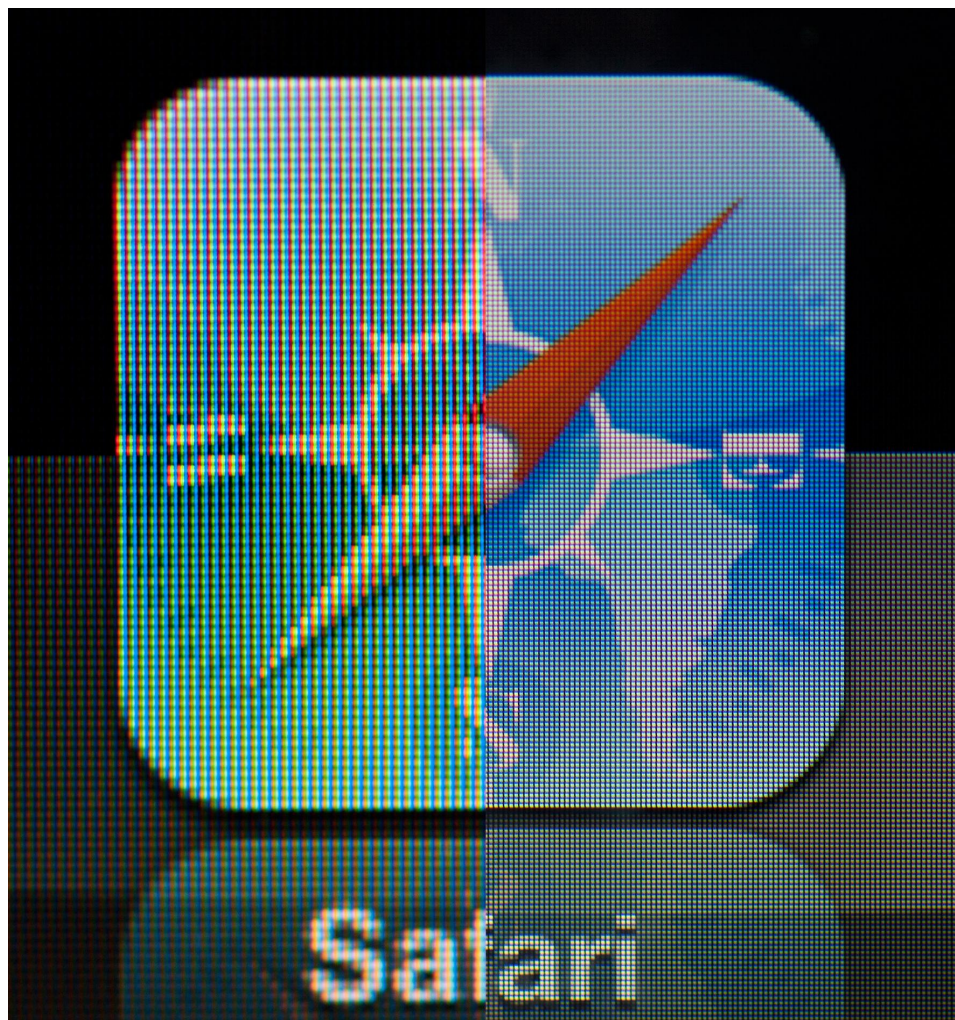
Down-  
sampled  
bitmap pixels

Standard



Retina





**Ретинизация**

# Растровая графика

Растровая графика — сетка разноцветных пикселей. Размер сетки фиксированный.

Растровые форматы: PNG, JPG, GIF и другие.



# Определение ретина-экрана в CSS

```
@media (min-resolution: 192dpi), (min-resolution: 2dppx) {...}
```

- dpi - dots per inch
- dppx - dots per pixel

# Векторная графика

Векторная графика — изображения, описываемые математическими функциями. Размер не важен. Бесконечно масштабируемы в любом направлении без потери качества.

# Общие правила

Текст — выглядит отлично сам по себе.

PNG / JPEG — увеличивайте изображения в  $n$  раз.

Фоновые изображения — используйте background-size.

Отдавайте разные версии растровой графики.

Оптимизируйте растровую графику.

SVG — выглядит отлично.

Фавиконки — подготовьте favicon.ico с иконками как минимум двух размеров.

# **Адаптивная графика**

**Резиновые изображения**  
**max-width: 100%**

# Предзагрузка изображений

Предзагрузка изображений запускается до завершения построения DOM.

Информация об адаптивной графике должна быть доступна на этапе построения DOM, то есть она должна находиться в HTML-коде.

[Bruce Lawson on Responsive Images](#)

# Адаптивный `img`

Реагировать на ретину — загружает разные картинки для разных плотностей пикселей.

Реагировать на ширину вьюпорта — загружает разные картинки для разных медиавыражений.

«Прогрессивно» работать с новыми форматами изображений.

Даже когда размеры изображения заданы относительно.

# Браузер выбирает картинку

Браузер знает:

- Ширина вьюпорта
- Ретиновость экрана

Браузер не знает:

- Фактические размеры картинки
- Размер холста относительно вьюпорта



# srcset

Тег `picture` в сочетании с новыми атрибутами определяет настройки, подменяя `src` у `img`, когда это необходимо. Он даёт возможность разработчику передать браузеру информацию, необходимую для выбора подходящей картинки.

Чтобы отдавать увеличенные картинки только на ретиновые экраны, используем `srcset`.

Дескриптор `x` - описывает плотность пикселей экрана

Дескриптор `w` - описывает реальные размеры изображения в пикселях

# Назначение `srcset`

Дать браузеру возможность выбрать лучший вариант из предложенных.

Рассказать браузеру о фактических размерах предложенных картинок.

Не предназначен для кадрирования, предложенные варианты — это одна и та же картинка разного качества.

# Векторная графика

# Плюсы и минусы

## Плюсы:

- Отзывчивость
- SVG — это код, легко редактируется руками
- Работает в браузере, можно прямо в HTML, включает стили и скрипты
- Меньший размер по сравнению с растровыми изображениями

## Минусы:

- Не поддерживается в старых браузерах
- Только простая графика

# Использование SVG

Через `img` - не оформляется

Как фон - не оформляется

Inline SVG в HTML - оформляется, не кэшируется, мусорит

SVG спрайт

# Спрайты

# Растровые спрайты

Растровый спрайт это большая картинка частями которой являются более мелкие картинки. Используется для уменьшения кол-ва ресурсов запрашиваемых с сервера. Обычно в спрайт объединяют оформительные изображения (иконки, плашки и т. п.)

# Векторный спрайт

Просто набор символов (symbol).

svg -> symbol