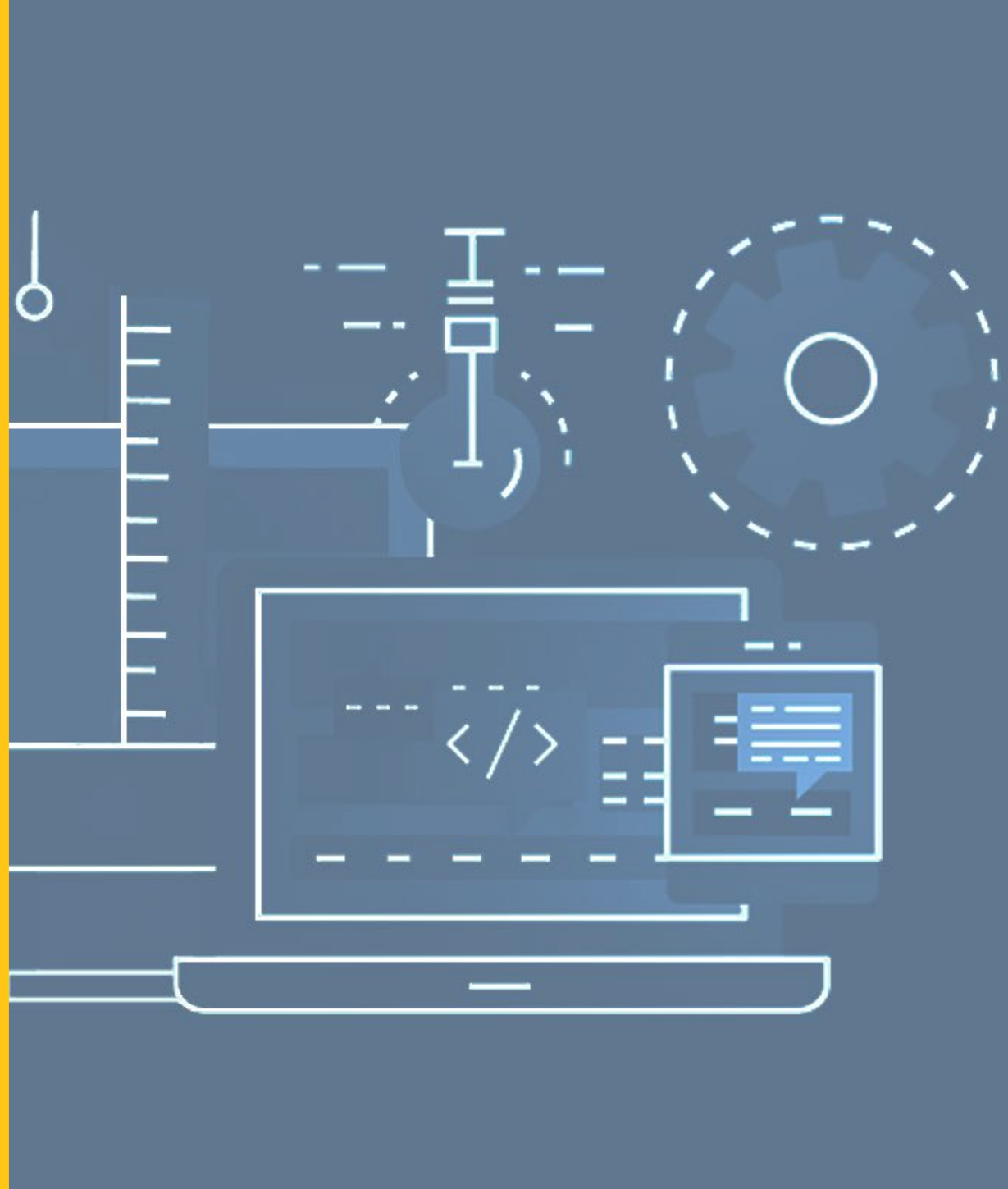
SHELLY GRAHAM, 01/21/2021

WEB DEV 3 WINTER 2021

Week 3:
BIG IMAGES & IMAGE OPTIMIZATION PART 1



WEEK 3:

IMAGE OPTIMIZATION

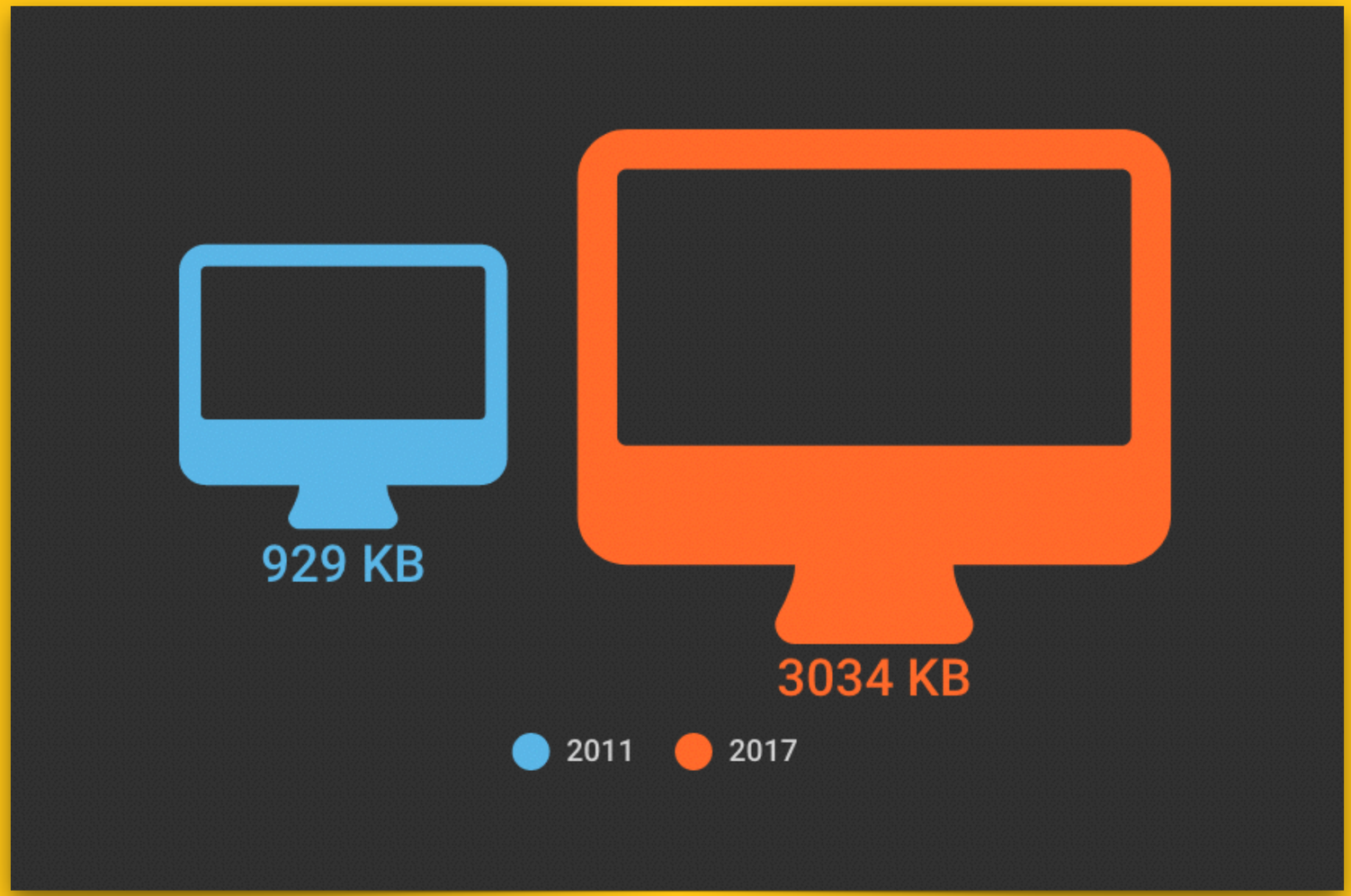
WHY IMAGE SIZE MATTERS

- The average website is about 3MB right now
- Used to be “just” 1MB back in 2012

- Use Sitechecker to check size of a website

- Check out HTTP Archive to see annual website reports

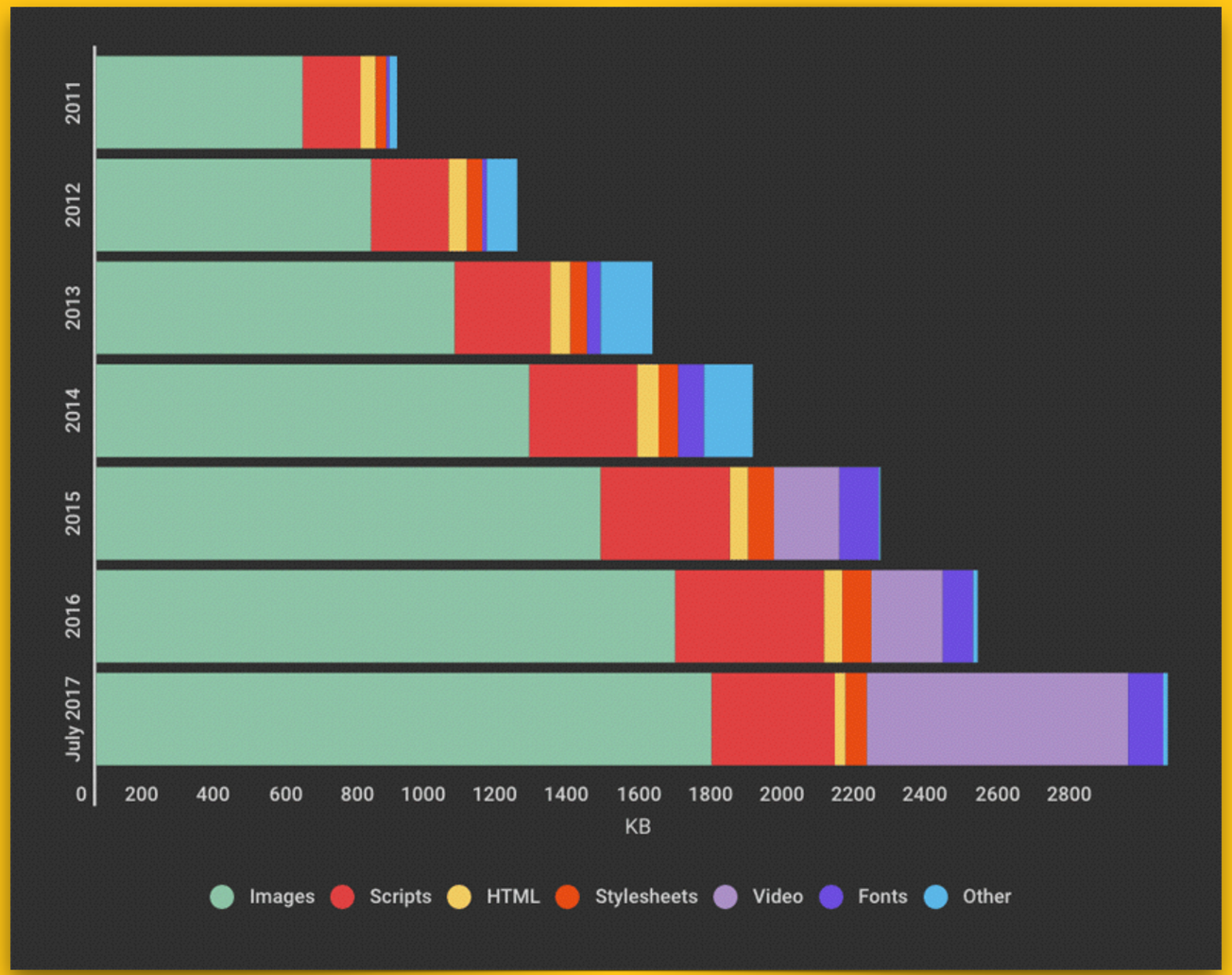
AVERAGE WEBSITE SIZE



IS THE SIZE OF A WEBPAGE A METRIC WE SHOULD CARE ABOUT?

- YES!!!
- In a world that is increasingly reliant on mobile phones you as a developer need to keep page load at a minimum!
- It's important to minimize bounce rate and keep interactions on page as high as possible!
- The easiest way to reduce the size of a website and hence make it faster is through IMAGES and VIDEOS!

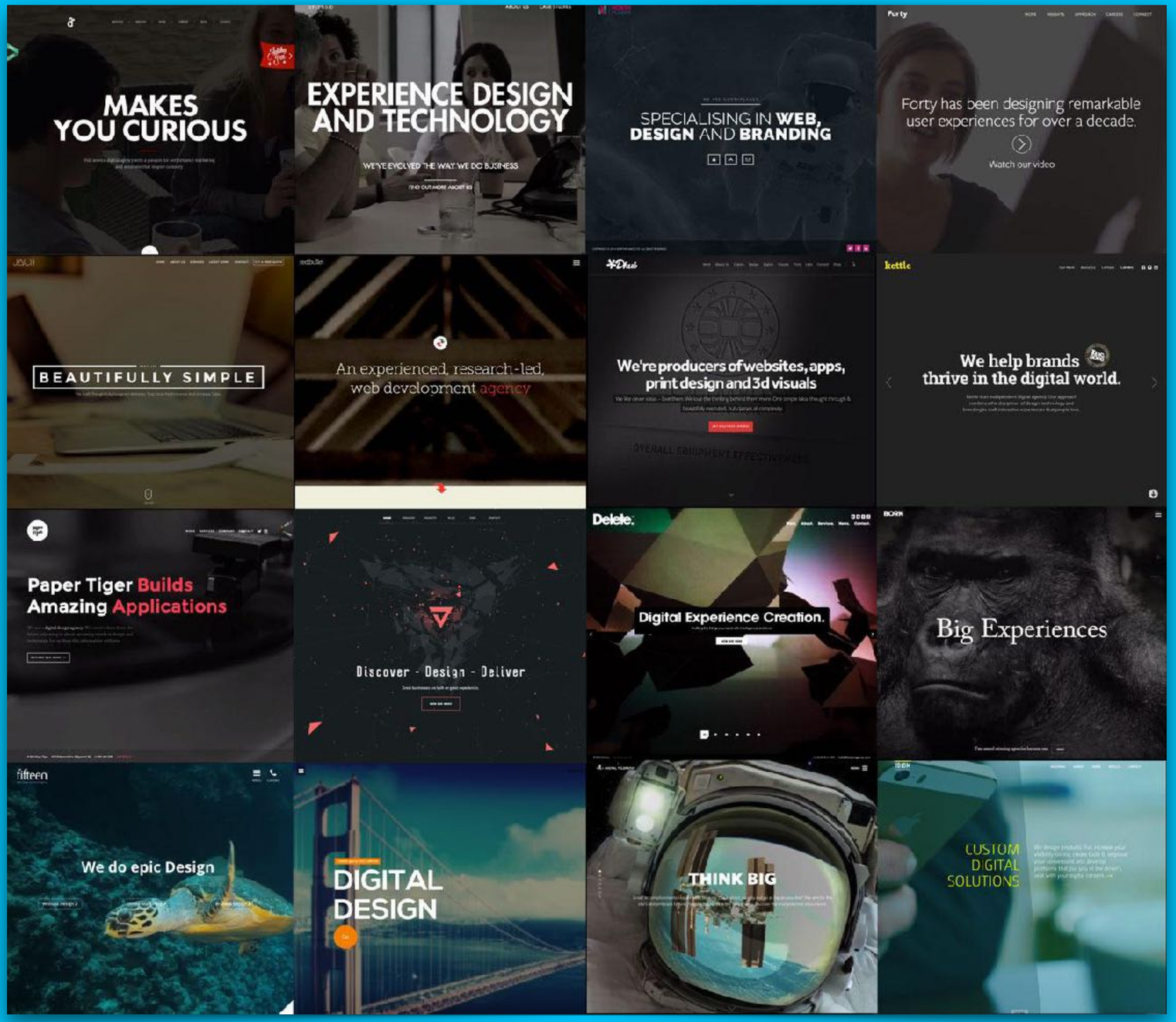
FILE USE/GROWTH OVER THE YEARS



THERE IS NO SUCH THING AS A TYPICAL WEBSITE!

THEY MIGHT LOOK THE SAME BUT CAN BE BUILD COMPLETELY DIFFERENT


BIG IMAGES LOOK GOOD...
...said every agency ever...



Literally every agency ever...

WHY DO SO MANY WEBSITES LOOK THE SAME

- In short: Generic content makes more money faster
- These types of websites are easy to style and can be created without the need for a developer —> Say Hello to Wix and Squarespace! -.-

- This website style is so mainstream that clients started asking for it!
- Creative agencies are often at fault because they have the creative freedom to design outside the box - but often won't!

- All Websites Look The Same
- Huge
- Brightlabs
- Hello Monday

ALL THESE BIG IMAGES TAKE UP SPACE!

- Large file sizes —> large websites —> long periods of loading —> annoyed users —> high bounce rate —> profit decline —> upset client/company —>

IMAGE OPTIMIZATION

The quickest way to improve page performance.

Maybe country performance as well.

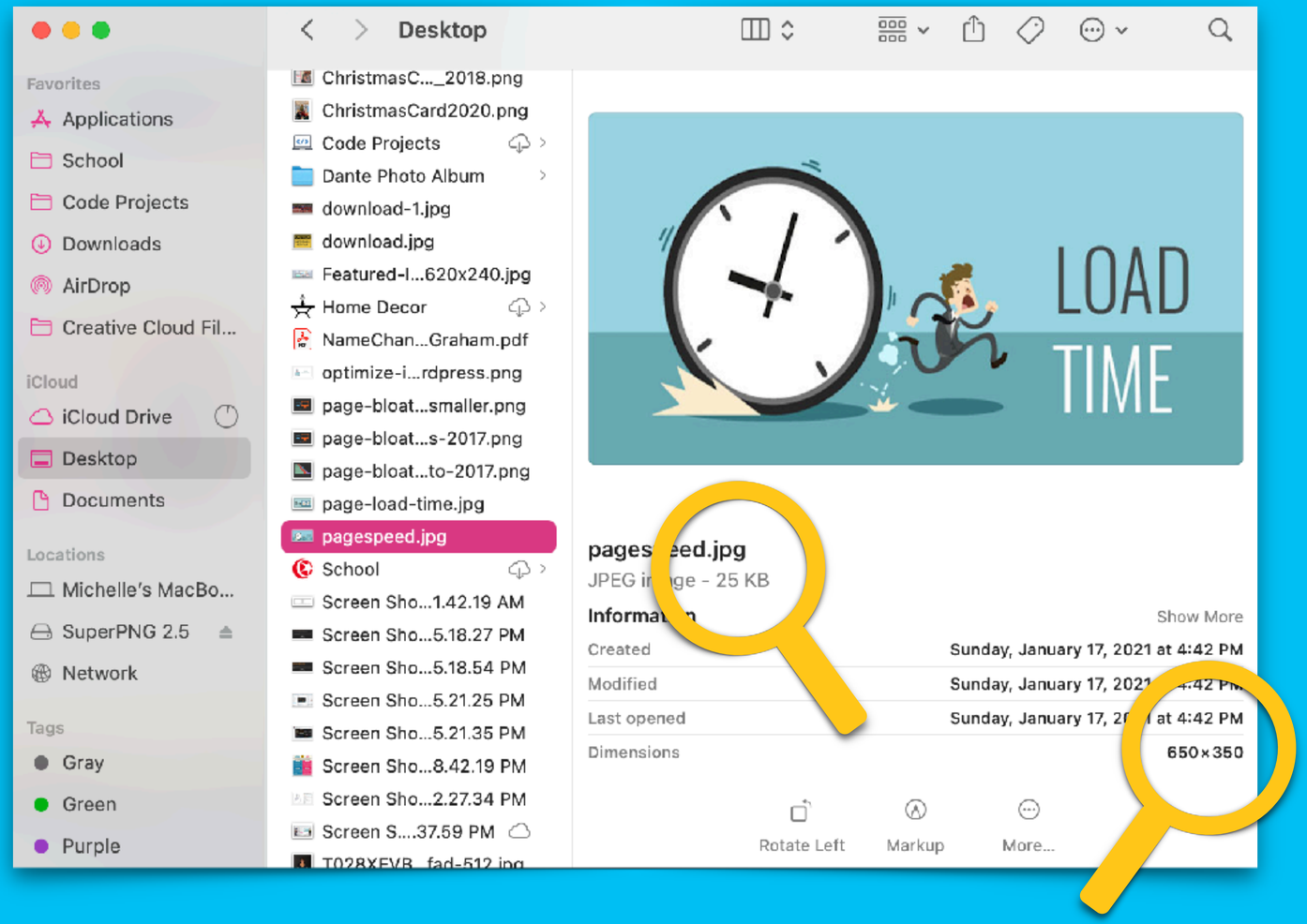
WHAT IS IMAGE OPTIMIZATION

- Process that contributes to faster web content load times
- Helps with SEO and Google ranking
- Helps with accessibility
- Should become second nature!

STEP 1:
FIND OUT IMAGE SIZE

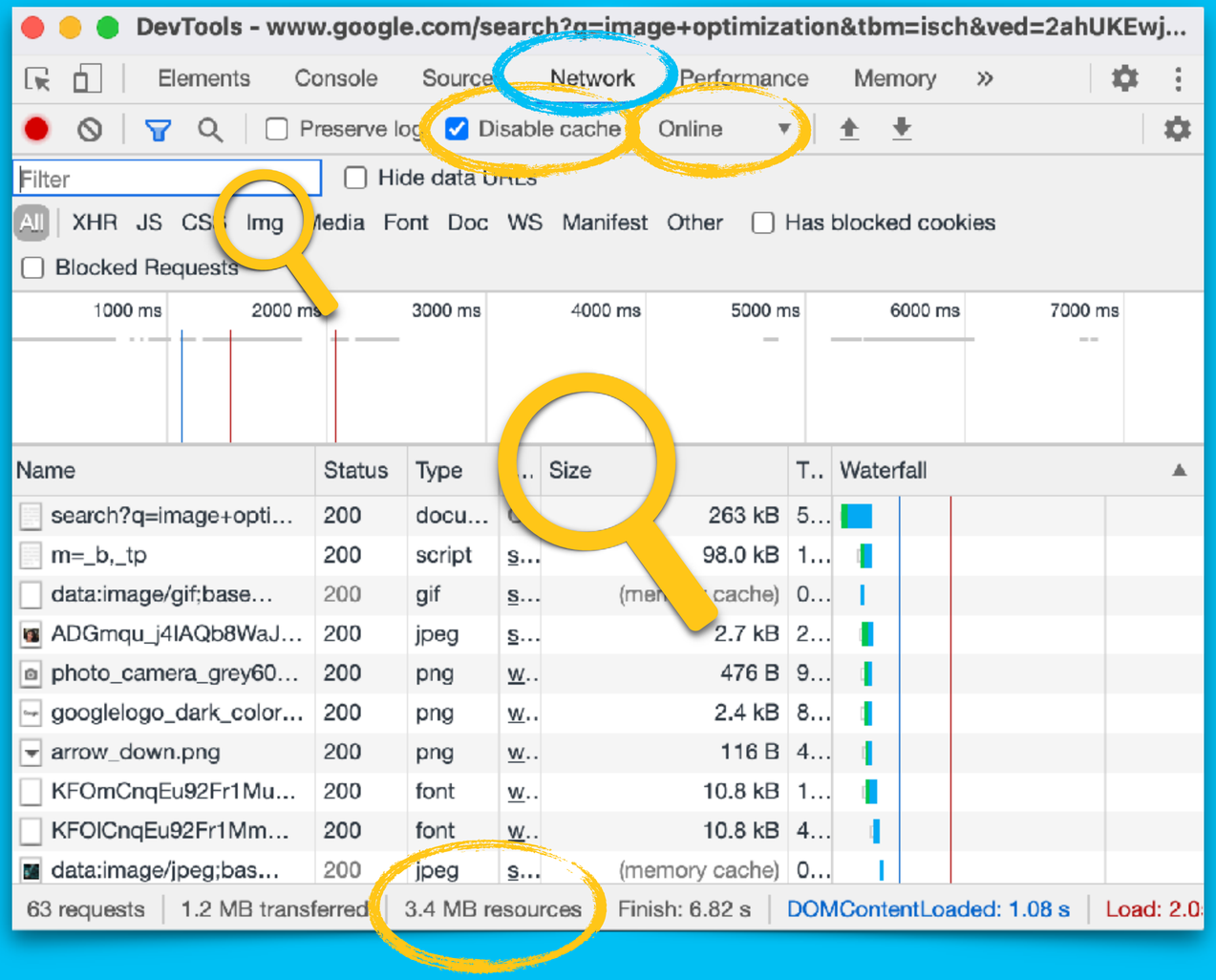
1. HOW TO FIND OUT IMAGE SIZE

- Via Computer:
  - Mac: List View Finder
  - Windows: List View in Explorer
  - Column view also shows image dimensions
- Photoshop: Image > Image Size



2. HOW TO FIND OUT IMAGE SIZE

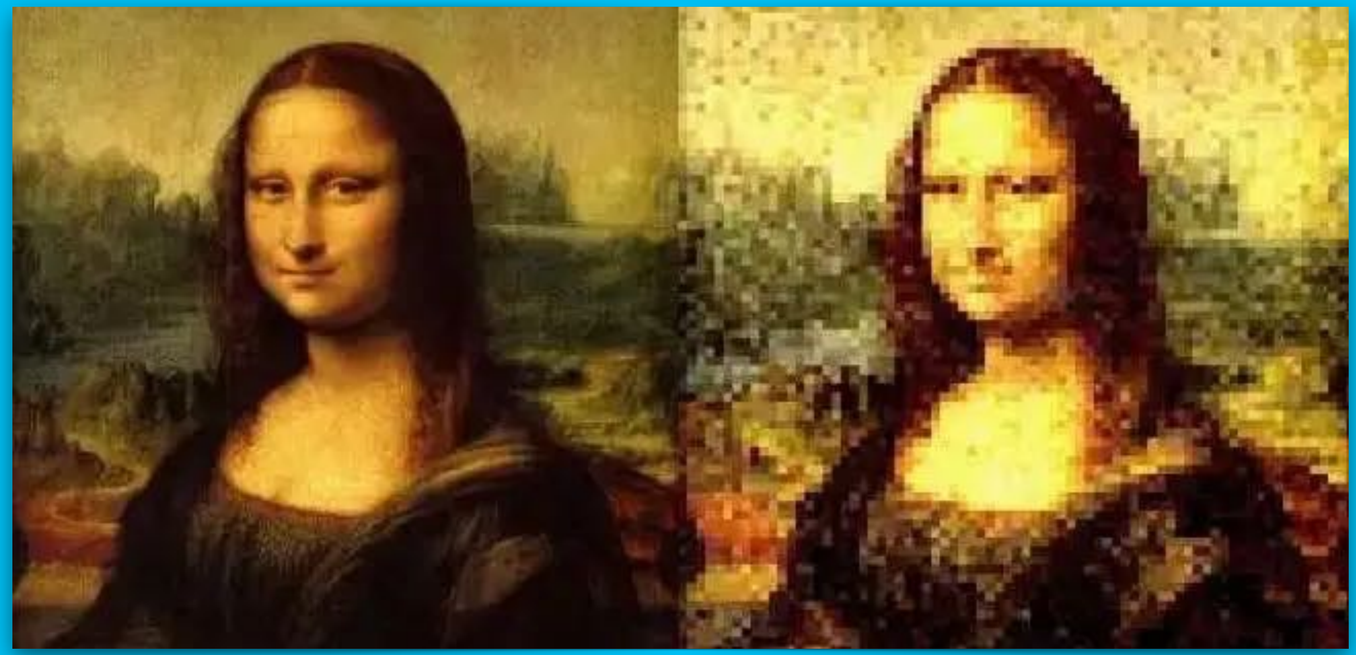
- Via Browser:
  - (Chrome) Dev Tools!
  - Network Tab!
    - Disable Cache
    - Online/Speed Status
    - Total size of resources



STEP 2:
FIND OUT IMAGE FORMAT

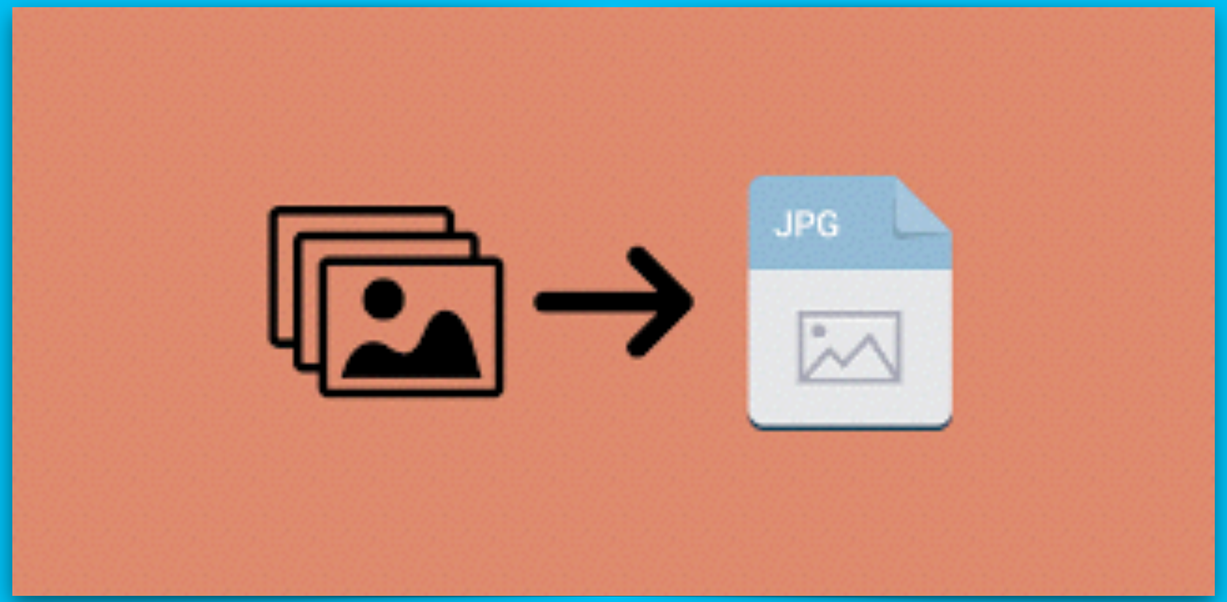
1. JPEG AKA JPG

- Joint Photographic Experts Group
- Early versions of Windows only allowed 3-letter file extensions which is why we have JPG
- Lossy Compression: Irreversible process of image compression with data loss due to inexact approximations



1. JPEG AKA JPG

- Often used for print
- Good for:
  - Photographs
  - Highly complex images
  - Images without transparency



2. GIF

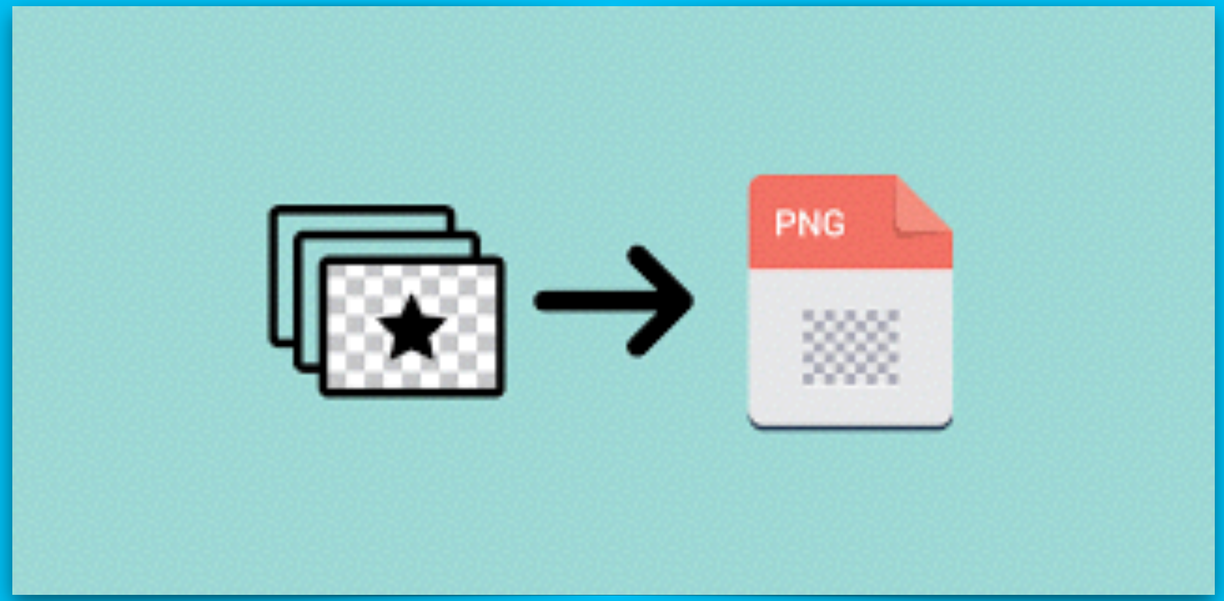
- Graphics Interchange Format
- Good for:
  - Nothing...
  - Ability to bundle multiple images into a single file for animation
  - But really nothing...

3. PNG

- Portable Network Graphic
- Created in 1996 to basically replace the GIF
- Mainly used for images on the internet
- Offers transparency! —> Logos!
- Supports lossless compression

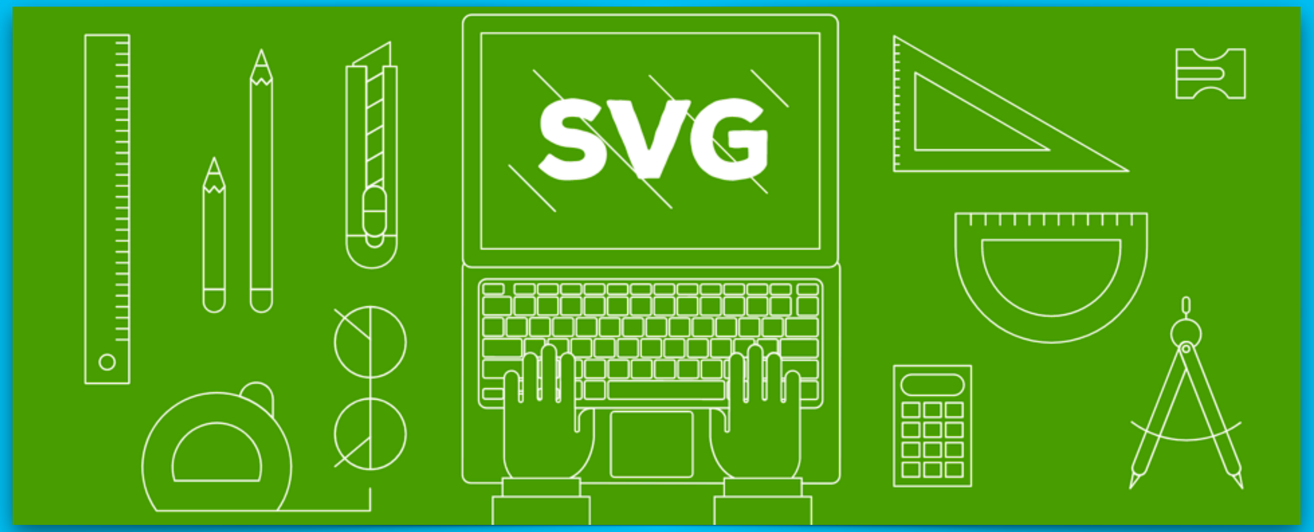
3. PNG

- Good for:
  - Illustrations (Photoshop/ Illustrator files)
  - Hard edges
  - Transparency



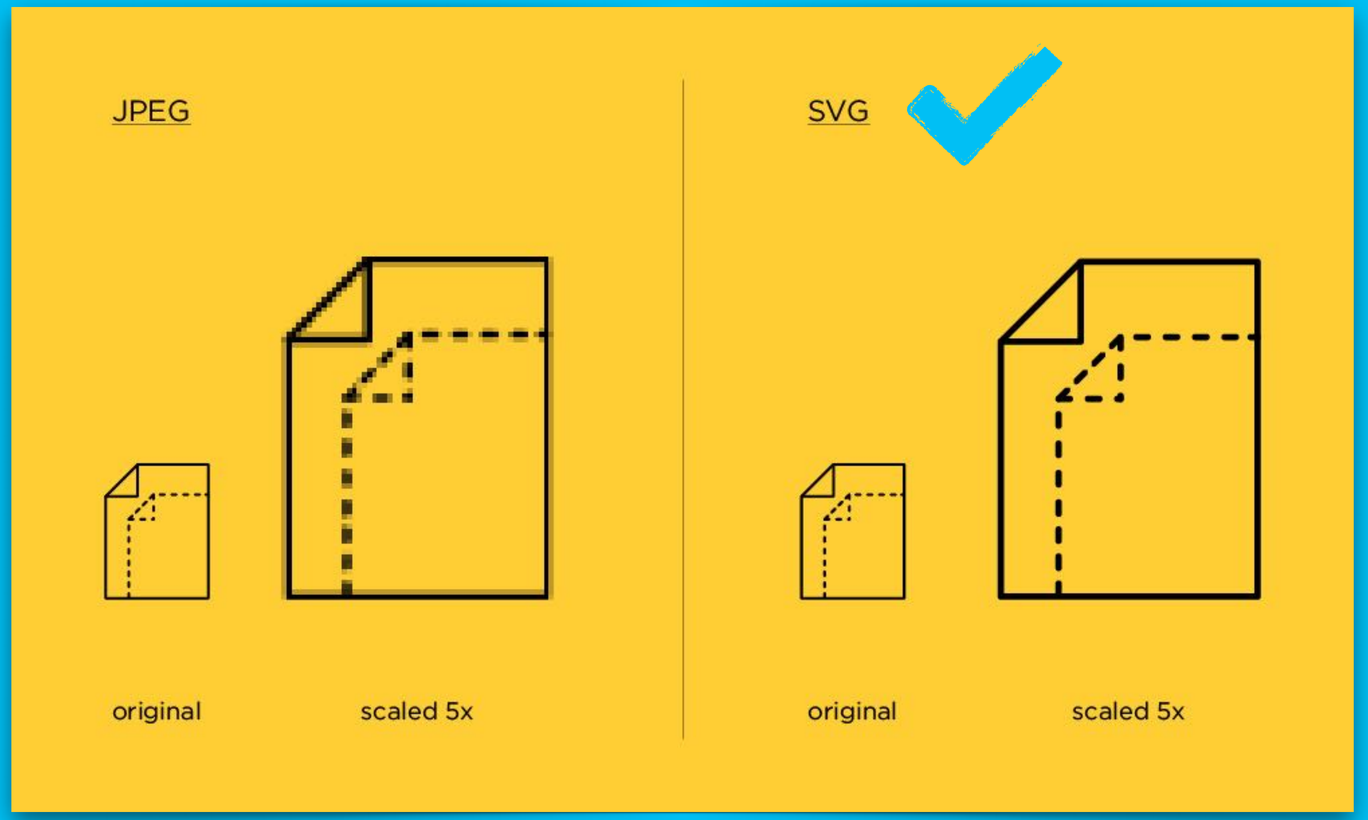
4. SVG

- Scalable Vector Graphic
- Doesn’t use pixels but coordinates, shapes and numbers
- Is indefinitely scalable
- Format that meets current web dev demands of scalability, responsiveness, interactivity, programmability, performance and accessibility the best!



4. SVG

- SVGs are text files, that describe lines, shapes, colors and text
- Embedded in HTML Code
- Can be changed via CSS and JS
- Smallest in size over JPG & PNG
- Good for:
  - Vector Art
  - Icons
  - Large, simple images



FUTURE IMAGE FORMATS

- WebP:
  - Googles alternative to JPG
  - Employing both loss and lossless compression!
  - Up to 80% smaller than JPGs and 45% smaller than PNGs
  - Growing browser support
- HEIF/HEIC:
  - High Efficiency Image File/Container
  - Support both still images and image sequences (burst photos and videos)
  - Non-destructive editing operations —> reverse changes whenever you want!
  - So far no browser support
- FLIF:
  - Free Lossless Image Format
  - Potentially the new PNG

Source

STEP 3:
IMAGE COMPRESSION

1. GENERAL COMPRESSION

- Always try to opt for lossless compression to maintain quality

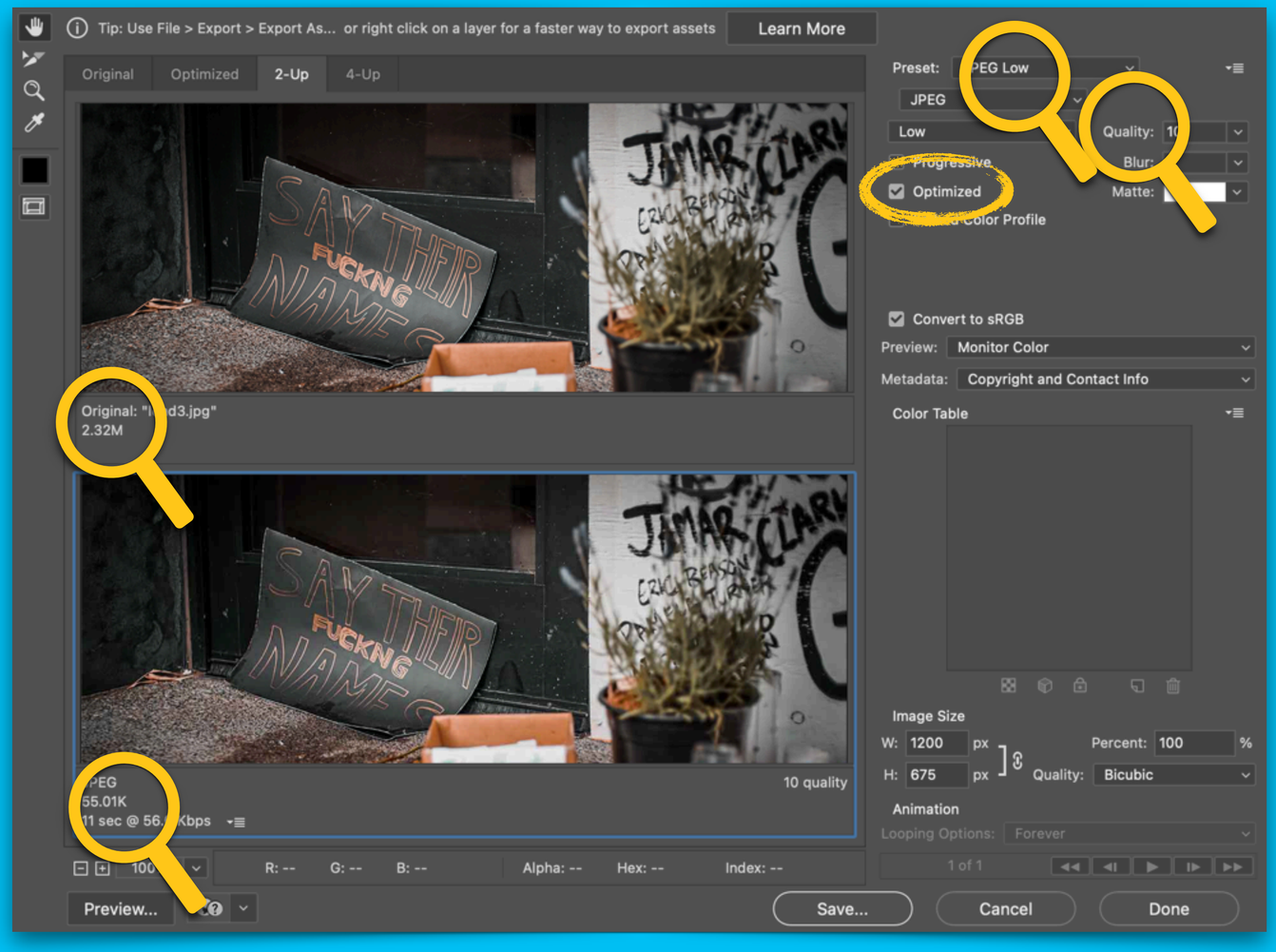
- GULP-IMAGEMIN
  - Image compression built into Circus Starter!
- Mac: ImageOptim (I prefer to use this on PNGs)
- Windows: FileOptimizer

2. PHYSICAL IMAGE SIZE

- Use images with the right dimensions
- You control the CSS, you know the maximum size of image you need!
- Designers will often hand you images that are too big or have super high resolution! —> make it smaller!
- Size images to their role on the page!
- Background images don’t have to be huge!
- Use lazy loading for images!

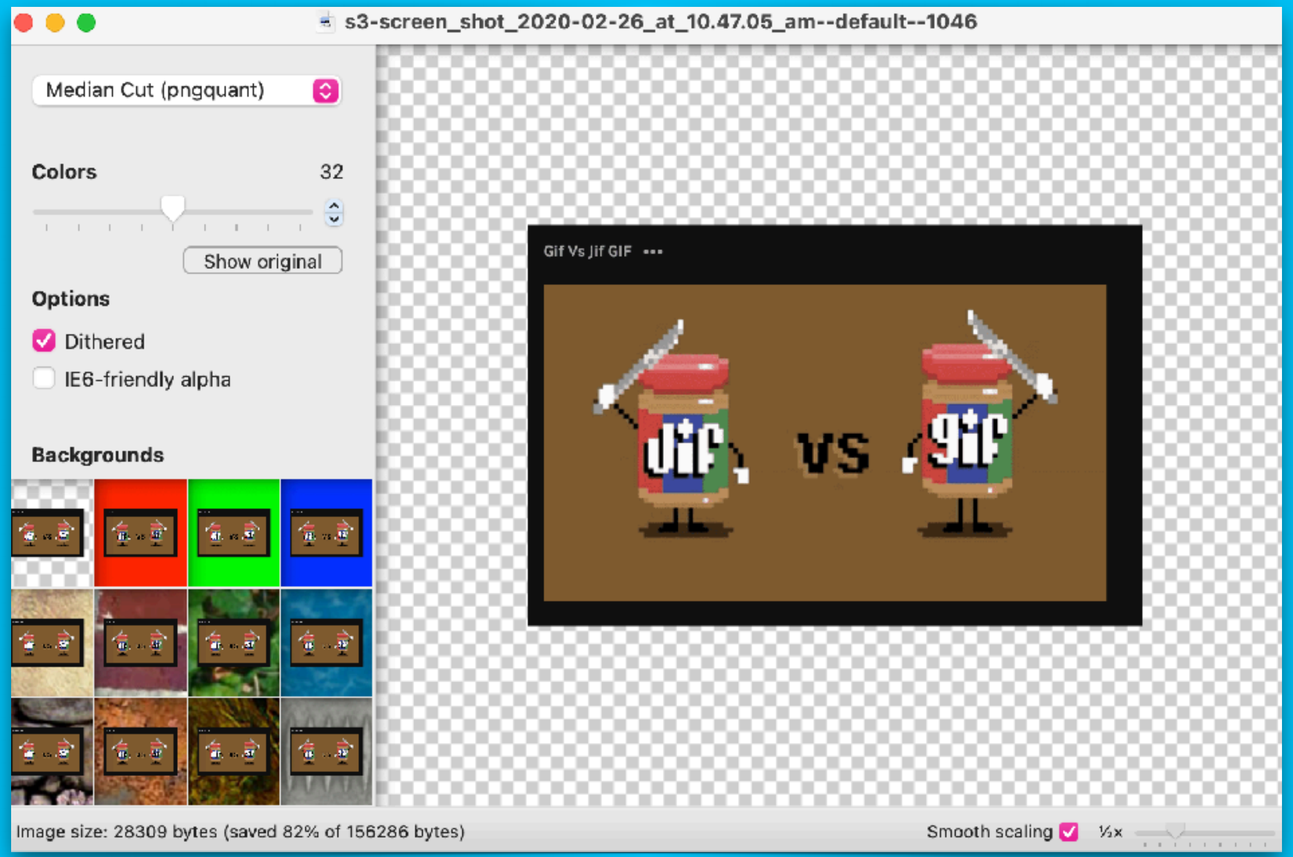
3. OPTIMIZE JPGS

- Photoshop: Save for Web
- You cannot beat this tool!
- Keep dimensions in mind
- Work with Info Window



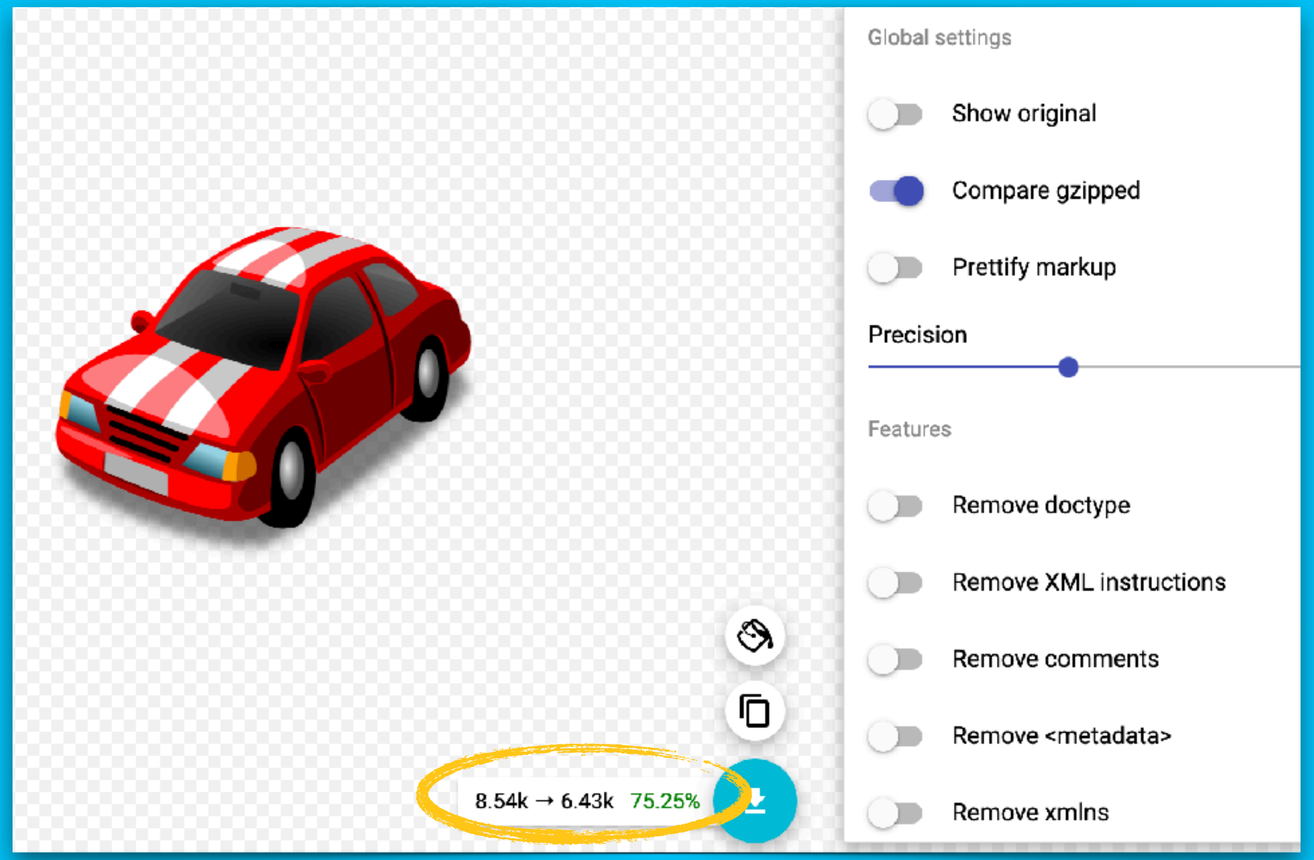
4. OPTIMIZE PNGS

- Also Photoshop
- Use Image Alpha tool
- Use ImageOptim/FileOptimizer tool



5. OPTIMIZE SVGS

- Talk to your designer!
- Use SVG OMG

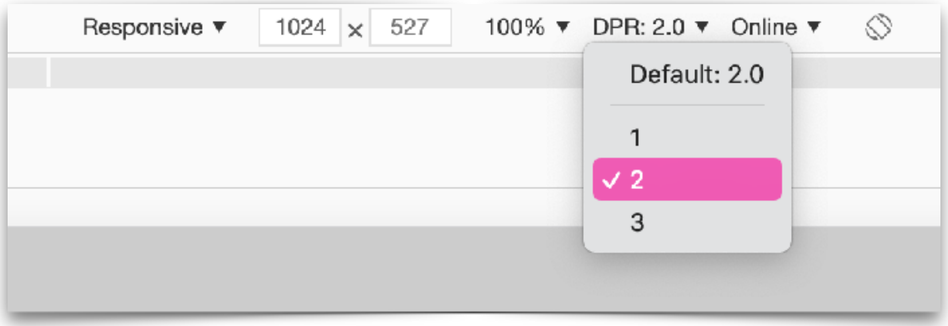


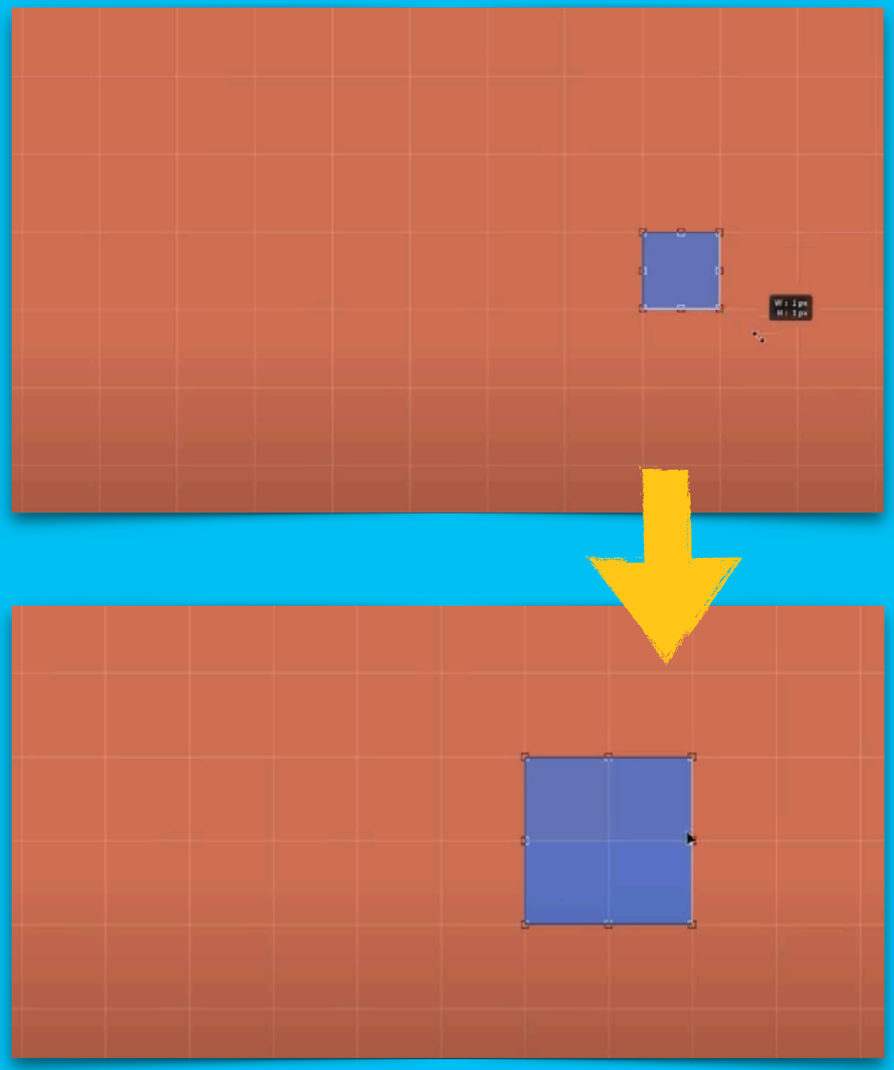
STEP 4:
MULTIPLE IMAGE SIZES

MORE THAN ONE SIZE?

- Presenting different images on different devices
- 1 pixel is not just 1 pixel anymore
- Retina displays came along and rendered 1px as 4px = 2X
- And they even introduced 3X: 1px rendered as 9px!
- We do this with the help of:
  - srcset attribute
  - sizes attribute
  - picture element

PIXEL RATIO AKA DENSITY





1. SRCSET

- srcset = source set
- Create multiple possible image sources in one image tag, separated by commas
- Is property of image element
- For resolution change only, not responsive images!
- Set regular image src to lowest quality image as fallback
- Can be used with pixel ration aka density or better width

Density

```
<img
    src="lawrence-ipsum-1x.jpg"
    srcset="lawrence-ipsum-1x.jpg 1x,
            lawrence-ipsum-2x.jpg 2x,
            lawrence-ipsum-3x.jpg 3x"
    alt="lawrence ipsum"
>
```

Width

```
<img
    src="lawrence-ipsum-1x.jpg"
    srcset="lawrence-ipsum-1x.jpg 567w,
            lawrence-ipsum-2x.jpg 987w,
            lawrence-ipsum-3x.jpg 1243w"
    alt="lawrence ipsum"
>
```

2. SIZES

- Uses media queries in HTML - woah!
- Yes, usually all styling in CSS file ONLY but this is the exception to the rule!
- Media queries are being executed before CSS is even loaded to determine which image to use!
- Can get messy though

```
<img
    src="lawrence-ipsum-1x.jpg"
    srcset="lawrence-ipsum-1x.jpg 567w,
            lawrence-ipsum-2x.jpg 987w,
            lawrence-ipsum-3x.jpg 1243w"
    sizes="
            (min-width: 560px) 50vw,
            (min-width: 798px) 75vw,
            (min-width: 960px) calc(100vw - 2em),
            100vw"
    alt="lawrence ipsum"
>
```

3. PICTURE

- Builds off of srcset
- Mainly used for art direction
- Use this for multiple image formats
- Doesn’t do much on it’s own, needs the img tag inside
- Browser has little control so use sparingly!


```
<picture>
    <source
        media="(min-width: 960px)"
        srcset="img/lawrence-ipsum__large.jpg 534w,
                img/lawrence-ipsum__large2X.jpg 1068w,
                img/lawrence-ipsum__large3X.jpg 1602w"
    >
    <source
        media="(min-width: 798px)"
        srcset="im
                                                    4w,
                                                    2w"
                ence-ipsum__small.jpg,
                img/lawrence-ipsum__small2X.jpg,
                img/lawrence-ipsum__small3X.jpg"
    >
    <img
        src="img/lawrence-ipsum__large.jpg"
        alt="lawrence ipsum"
    >
</picture>
```

3. PICTURE

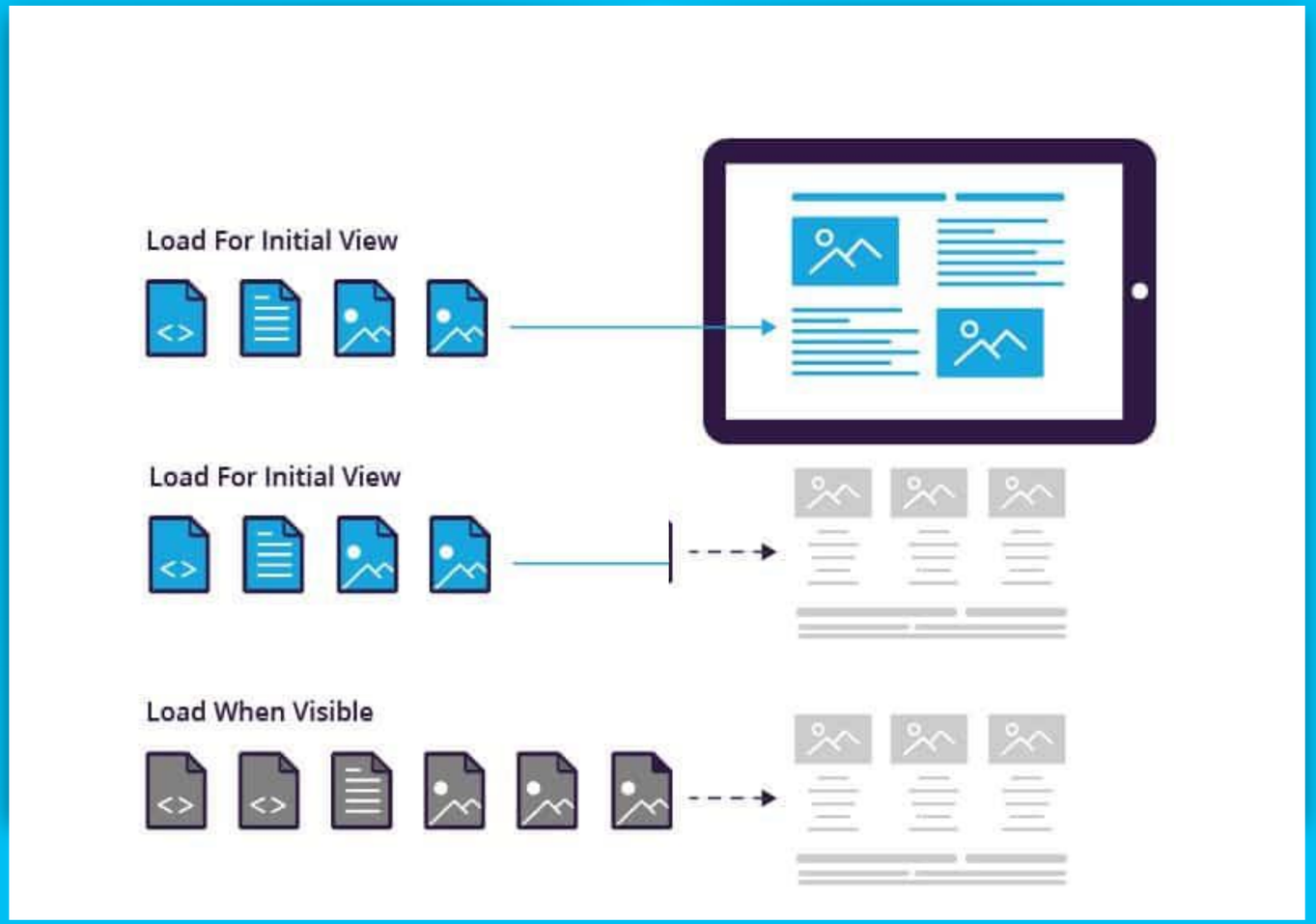
- Can’t use src in source so you use srcset!
- The source tag, with help from src attribute determines which image to load in the img tag! So only 1 images gets loaded!
- Order is important! Go from large to small and have fallback image at bottom
- Only style the fallback img element, not the picture tag itself!

```
<picture>
    <source
        media="(min-width: 960px)"
        srcset="img/lawrence-ipsum__large.jpg 534w,
                img/lawrence-ipsum__large2X.jpg 1068w,
                img/lawrence-ipsum__large3X.jpg 1602w"
    >
    <source
        media="(min-width: 798px)"
        srcset="img/lawrence-ipsum__medium.jpg 474w,
                img/lawrence-ipsum__medium2X.jpg 984w,
                img/lawrence-ipsum__medium3X.jpg 1422w"
    >
    <source
        media="(min-width: 560px)"
        srcset="img/lawrence-ipsum__small.jpg,
                img/lawrence-ipsum__small2X.jpg,
                img/lawrence-ipsum__small3X.jpg"
    >
    <img
        src="img/lawrence-ipsum__large.jpg"
        alt="lawrence ipsum"
    >
</picture>
```

STEP 5:
HOW TO USE COMPRESSED IMAGES

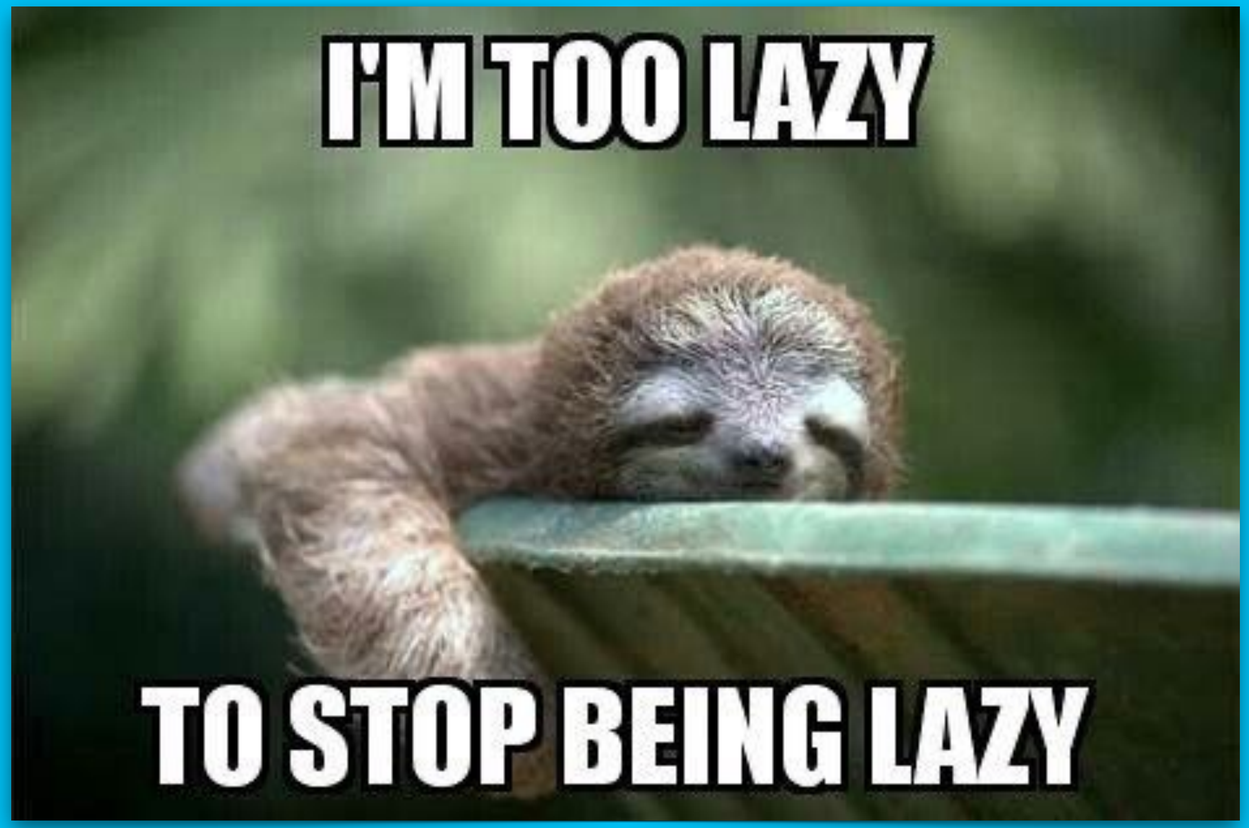
1. LAZY LOADING

- Instead of bulk loading all the content when the page is accessed, content is only loaded when user accesses a part of the page that requires the content!
- Above the fold vs. Below the fold
- Lazy Loading for multiple important purposes:
  - Page Speed
  - Accessibility
  - SEO



1. WHY USE LAZY LOADING

- For any website, loading the DOM is crucial!
- We place scripts at the end of body to load them after DOM has fully loaded
- If we have to wait on images to load, all other components have to wait as well
- Helps with speed and bandwidth —> crucial for mobil users!



1. HOW TO LAZY LOAD

- All roads lead to Rome!
- Used to be a lot more complicated —> Intersection Observer API
- Nwo it’s super easy!

1. Native Lazy Loading
   - Is used directly on image in HTML!
   - Simply add the loading attribute to your image, picture, or iframe element
   - Mostly supported across browsers

```
<img
    src="/src/img/lauren-ipsum.jpg"
    alt="Lauren Ipsum"
    loading="lazy"
>
```

- loading=“eager” is default
- loading=“lazy”

2. HOW I LIKE TO LAZY LOAD

- Follow Craig Bucklers approach: https://codepen.io/craigbuckler/full/yPqLXW
- Add link and script to HTML file
- Create smaller image with class="preview"
- Surround image with a tag and add class="progressive replace"
- The preview and full-size images must have the same aspect ratio

2. HOW I LIKE TO LAZY LOAD

- Add data-src attribute
  - src attribute on image element will trigger browser to load image
  - data-src will defer the load!
  - Image still needs a src attribute, if you don't use it this won't work!
  - Can be combined with lazy loading

```
<img
    src="/src/img/load2.jpg"
    alt="load 2"
    data-src="/src/img/load2.jpg"
>
```

3. SPRITE SHEETS

- Sprite sheets already exist since the first days of computer games, traditionally used for animation
- Create one big image that contains many other images instead of dealing with all the images individually
- Simply use background image and background position to show certain area of sprite sheet
- Use SpriteCow to help with background positioning!

HTML

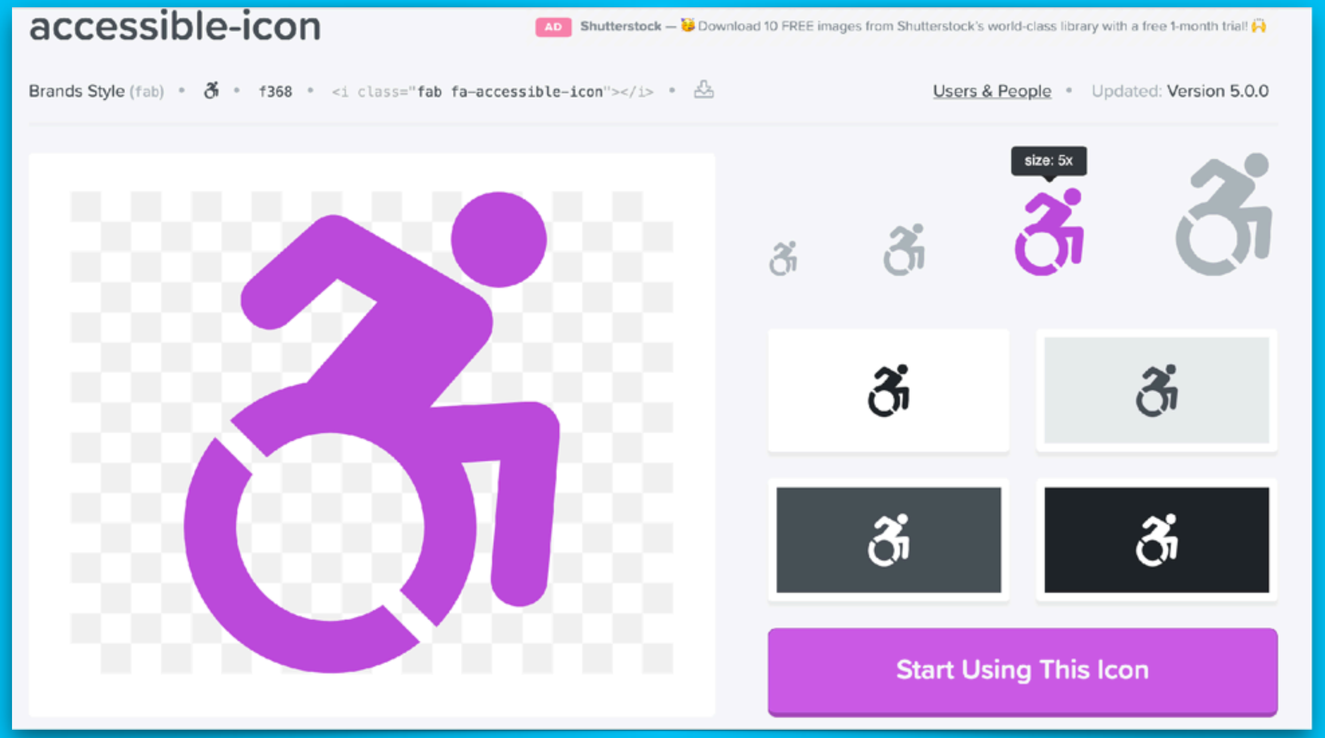
```
<ul class="tut-nav">
    <li class="first"><a class="drupal" href="#">Drupal</a></li>
    <li><a class="sass" href="#">Sass</a></li>
    <li><a class="compass" href="#">Compass</a></li>
    <li class="last"><a class="ee" href="#">Expression Engine</a></li>
</ul>
```

CSS

```
.tut-nav li {
    list-style: none;
    margin:20px 0;
}
.tut-nav li a {
    background: url(../img/logosprite.png) no-repeat;
    display: block;
    padding: 28px 10px 28px 80px;
}
.tut-nav li a.ee {
    background-position: 0 -134px;
}
.tut-nav li a.compass {
    background-position: 0 -270px;
}
.tut-nav li a.sass {
    background-position: 0 -408px;
}
```

4. ICONS

- Font Awesome: font and icon toolkit based on CSS
- Amazing for so many different things!
- Use them in your nav, in your footer, in your text. Really wherever you want!
- Use in HTML or CSS



WHAT'S THE BEST IMAGE FORMAT TO "SAVE AS"?

Isovera

What is the final image's use case?

DIGITAL

What are you saving?

PHOTOS

Does the image need to be edited?

NO

JPEG

Best for saving photos or lifelike images with a small file.

Best uses:
- Storing images in a small file
- Saving images and photos online
- Images from and for the web should be generally saved as JPEGs

Unfavorable uses:
- When maximum quality is required
- For editing and re-saving images

YES

GRAPHICS

PRINT

Does the image need to be edited?

YES

Do images need to offer transparent backgrounds?

DOESN'T MATTER

SVG

Best for saving graphics for scaling up or down digitally.

Best uses:
- Animation
- To create, edit, or save logos, icons, or shapes
- When images and graphics will be featured on a variety of screen sizes, or on retina screens

Unfavorable uses:
- Working with small photos
- When you need transparent backgrounds

Is your file larger than 2GB?

NO

YES

NO

YES

Do images need to offer transparent backgrounds?

YES

Does image layers need to be preserved?

NO

PNG

Best for saving transparent background graphics.

Best uses:
- Saving icons, logos, graphs, diagrams, or infographics (like this one!)
- When you need transparent backgrounds or graphics with a variety of backgrounds

Unfavorable uses:
- Posting images online (PNG files are large)
- Widespread use (due to lack of support)
- When large file size is an issue

NO

YES

YES

TIFF

Best for creating and saving images and graphics for print use.

Best use cases:
- Creating and editing files
- Saving files when the file is >2GB
- Creating print materials or graphics to be used offline

Unfavorable use cases:
- Digital use
- When large file size is an issue

PSD

Best for creating and saving images and graphics with layers.

Best uses:
- Keeping "trails" of image manipulation (layers)
- Editing and altering an image
- Sharing in-progress edits

Unfavorable uses:
- Storing and sharing final images and graphics
- Sharing with folks who don't have PS
- When large file size is an issue

See more at www.isovera.com

QUESTIONS?

HOMEWORK

- Optimize the included site to have a total load weight below 700k: https://github.com/michellejames/wd3_week3_homework-demo
- The site as it is has a final load of 20.3MB. Using what we discussed in class, optimize the images on the site to bring the final load below 700KB. More importantly, make sure the images you end up with still look good at all reasonable browser sizes.
- Switch out social media icons with FontAwesome Icons.
- Bonus points if you are able to lazy-load one background-image and are able to get page load under 500KB.
- Feel free to change up the design and make it more you.
- Submit a link to the Git repo for this assignment.

FIN