Getting the LEAST from your images
Images are awesome
CONTRARY TO POPULAR BELIEF

I HAVE NO IDEA WHAT I AM DOING
But they aren’t great for web performance
in fact....
They are greedy

JavaScript

Images

Stylesheets, HTML, etc
Images are the biggest part of a web page.
The average web page makes 56 image requests per page

*Source: HttpArchive.org*
42% of all websites make over 50 image requests per page

*Source: HttpArchive.org*
The Web wouldn’t be the same without images
...So
We need to
Make more efficient use of requests
We need to

Make more efficient use of requests &

Reduce the size of the images
We will look at
We will look at

- Well known methods for reducing the size of your images and http requests
We will look at

- Well known methods for reducing the size of your images and http requests
- Some lesser known methods
We will look at:

- Well known methods for reducing the size of your images and http requests
- Some lesser known methods
- The future of images on the web
The Basics
Things everyone should be doing
Things everyone should be doing
Things everyone should be doing

Image optimization
Things everyone should be doing

Image optimization
Spriting images
Things everyone should be doing

Image optimization
Spriting images
Lazy loading
Image Optimization tools
Offline tools
- Image Magick
- Jpeg Optim
- Jpeg Tran
- Image Magick
- Jpeg Optim
- Jpeg Tran

- Image Magick
- Png Crush
- Png Out

Jpeg

Png
Online tools
Online tools

smushit.com
kraken.io
Automate it!
Optimisation APIs
Optimisation APIs

- kraken.io
- punypng.com/api
- webresizer.com
Sprites
Take lots of little images
..and combine into one image
.norwegian_flag
{
    width: 120px;
    height: 75px;
    background:url('/images/sprites/flag-sprite.png');
    background-position: -240px -84px;
}
1 request
1 request
many images
Recommendations
- No whitespace
Recommendations

- No whitespace
- Optimize & cache
Recommendations
- No whitespace
- Optimize & cache
- Aim for similar palettes
Lazy Loading
Lazy Loading

- Only load what the user sees
Lazy Loading

- Only load what the user sees
- Load images as the viewport moves
Lazy Loading

- Only load what the user sees
- Load images as the viewport moves
- Reduce unnecessary HTTP requests
Loads every image
12x Requests
3x Requests
Benefits

Reduced HTTP requests
Benefits

Reduced HTTP requests

A lot of users don’t scroll entire page
Benefits

Reduced HTTP requests
A lot of users don’t scroll entire page
Faster initial page load times
Moving on..
WebP Images
WebP is a new image format.
WebP is a new image format
They’re great because:

26% smaller than PNGs
They’re great because:

26% smaller than PNGs

25-35% smaller than JPGs
They’re great because:

- 26% smaller than PNGs
- 25-35% smaller than JPGs
- Image quality isn’t affected
Which means smaller web pages!
83%
WebP has its downsides
WebP has its downsides.

No IE support.
No FireFox support
No IE support
WebP has it’s downsides
WebP has its downsides

No iOS support
No Firefox support
No IE support

WebP has its downsides
Support is growing!
How can I use it?
Use it today

WebP converter tool

https://developers.google.com/speed/webp/
Use the WebP tool

```
C:\Users\d.hume>cwebp input.png -q 80 -o output.webp
```
Use the WebP tool

```bash
C:\Users\d.hume>cwebp input.png -q 80 -o output.webp
```

Specify compression

Specify sharpness & size
Use the WebP tool

Specify compression
Specify sharpness & size
Available for Win, Mac & Linux
<img src="cat.webp" />
JavaScript Shim

https://github.com/antimatter15/weppy
JavaScript Shim
https://github.com/antimatter15/weppy

Client Side
JavaScript Shim
https://github.com/antimatter15/weppy

Client Side

Server Side
http://tinyurl.com/BlogWebP
Give it a go!

tinyurl.com/LeastWebP
Compressive Images
Compressive Concept..
Compressive Concept..
Compressive Concept..
Compressive Concept..
Compressive Concept..
Compressive Concept..
Compressive Concept..
Compressive Concept..
YDN says...

“Don't Scale Images in HTML. Don't use a bigger image than you need just because you can set the width and height in HTML. If you need:

```html
<img width="100" height="100" src="mycat.jpg" alt="My Cat" />
```

then your image (mycat.jpg) should be 100x100px rather than a scaled down 500x500px image.”
Google says..

“Specify dimensions that match those of the images themselves. Don't use width and height specifications to scale images on the fly. If an image file is actually 60 x 60 pixels, don't set the dimensions to 30 x 30 in the HTML or CSS. If the image needs to be smaller, scale it in an image editor and set its dimensions to match”
Rationale

1. You don't want to download more bytes than necessary
Rationale

2. Browsers are bad at resizing images
Don’t download more bytes
Don’t download more bytes

Base resolution
(300 x 200 px)

Retina resolution
(600 x 400 px)

Jpg compression 80 / 21 kb

Jpg compression 31 / 16 kb (75% of base)

* From Daan Jobsis’ wonderful article @ http://tinyurl.com/retinarev
Don’t download more bytes

Big image low quality = GOOD

Small image high quality = BAD
Browsers are bad at resizing
Browsers are bad at resizing

150 images:

50 x 24bit pngs
50 x jpeg
50 x 8bit gif

One page: images displayed at 100%

* From Roy Tanck’s interesting study @ http://tinyurl.com/browserscaling
Browsers are bad at resizing

150 images:

- 50 x 24bit pngs
- 50 x jpeg
- 50 x 8bit gif

One page:
images displayed at 100%

Duplicate page:
images scaled to 61.5%

* From Roy Tanck’s interesting study @ http://tinyurl.com/browserscaling
Browsers are bad at resizing

No difference whatsoever

* From Roy Tanck’s interesting study @ http://tinyurl.com/browserscaling
Is it good?
Is it good?
Support for Compressive Images Chrome
Support for Compressive Images
All of them Chrome
Support for Compressive Images

All of them

Chrome

Opera
Support for Compressive Images

All of them

All of them

Chrome

Opera
Support for Compressive Images

All of them

All of them

Chrome

Opera

Firefox
Support for Compressive Images

All of them

All of them

All of them

Chrome Opera Firefox
Support for Compressive Images

All of them
All of them
All of them

Chrome
Opera
Firefox
IE
Support for Compressive Images

All of them

All of them

All of them

All of them

All of them

All of them

All of them

All of them

All of them

All of them

All of them

FireFox

Opera

Chrome

IE
Support for Compressive Images
All of them
All of them
All of them
All of them
Chrome
Opera
Firefox
IE

* at least I think so...
Got `<img>`?

Got Compressive
Give it a go!

tinyurl.com/LeastCompressive
Data URIs
“The fastest HTTP request is the one not made.”
Every image in a web page is a new HTTP request
Data URIs are different
<img src="cat.png" alt="Funny Cat" />
Data URIs are embedded as a Base64 encoded string.
<img src="data:image/png;base64,iVBORw0KGgoAAElEQVQI12P4//8/w38U5ErkJggg==" alt="Funny Cat"/>
Data URIs
Embedded in a web page
Data URIs

Embedded in a web page

No HTTP request is made
Data URIs

Embedded in a web page

No HTTP request is made

Faster load times
Data URIs
Embedded in a web page
No HTTP request is made
Faster load times*  

*Sometimes
Support is great

So....what’s the catch?
- Some browsers only support Data URIs up to a certain size
- Some browsers only support Data URIs up to a certain size
- Base64 encoded strings are actually around a third larger than the original image size
- Some browsers only support Data URIs up to a certain size
- Base64 encoded strings are actually around a third larger than the original image
- Need to re-encode on every change
- Some browsers only support Data URIs up to a certain size
- Base64 encoded strings are actually around a third larger than the original image
- Need to re-encode on every change
- Consume battery and CPU on mobile devices
Don’t worry!
Use it for:
- Images less than 32K
- Great for icons
Give it a go!

tinyurl.com/LeastDataURL
The Future
The Future?
The right image for the user
The right image for the user
srcset

HTML attribute
<img alt="my cat" src="cat.jpg" />
<img alt="My Cat" src="cat.jpeg" srcset="cat-HD.jpeg 2x, cat-mob.jpeg 100w, cat-mob-HD.jpeg 100w 2x" / >
<img alt="My Cat" src="cat.jpeg" srcset="cat-HD.jpeg 2x, cat-mob.jpeg 100w, cat-mob-HD.jpeg 100w 2x" />
<img alt="My Cat" src="cat.jpeg" srcset="cat-HD.jpeg 2x, cat-mob.jpeg 100w, cat-mob-HD.jpeg 100w 2x" />
picture

HTML Element
<img alt="my cat" src="cat.jpg" />
<picture width="500" height="500" id="pictureElement">
  <source media="(min-width: 45em)" src="tiger.jpg">
  <source media="(min-width: 18em)" src="cat.jpg">
  <source src="kitten.jpg">
  <img src="kitten.jpg" alt="cutekittyawww" lazyload>
<p>Lookit the cuuute kitty awwwww</p>
</picture>
Lookit the cuuute kitty awwwww
<picture width="500" height="500" id="pictureElement">
  <source media="(min-width: 45em)" src="tiger.jpg">
  <source media="(min-width: 18em)" src="cat.jpg">
  <source src="kitten.jpg">
  <img src="kitten.jpg" alt="cutekittyawww" lazyload>
<p>Lookit the cuuute kitty awwwww</p>
</picture>
<picture width="500" height="500" id="pictureElement">
    <source media="(min-width: 45em)" src="tiger.jpg">
    <source media="(min-width: 18em)" src="cat.jpg">
    <source src="kitten.jpg">
    <img src="kitten.jpg" alt="cutekittyawwww" lazyload>
    <p>Lookit the cuuute kitty awww</p>
</picture>
<picture width="500" height="500" id="pictureElement">
  <source media="(min-width: 45em)" src="tiger.jpg">
  <source media="(min-width: 18em)" src="cat.jpg">
  <source src="kitten.jpg">
  <img src="kitten.jpg" alt="cutekittyawww" lazyload>
  <p>Lookit the cuuute kitty awwww</p>
</picture>
picture + srcset
Lookit the cuuute kitty awwww
Lookit the cuuute kitty awwwww
Lookit the cuuute kitty awwww
<picture width="500" height="500" id="pictureElement">
  <source media="(min-width: 45em)" srcset="tiger.jpg 1x, tiger-HD.jpg 2x">
  <source media="(min-width: 18em)" srcset="cat.jpg 1x, cat-HD.jpg 2x">
  <source srcset="kitten.jpg 1x, kitten-HD.jpg 2x">
  <img src="kitten.jpg" alt="cutekittyawww" lazyload>
</picture>
<p>Lookit the cuuute kitty awwww</p>
Lookit the cuuute kitty awwww
<picture width="500" height="500" id="pictureElement">
  <source media="(min-width: 45em)" srcset="tiger.jpg 1x, tiger-HD.jpg 2x">
    <source media="(min-width: 18em)" srcset="cat.jpg 1x, cat-HD.jpg 2x">
    <source srcset="kitten.jpg 1x, kitten-HD.jpg 2x">
  <img src="kitten.jpg" alt="cutekittyawww" lazyload>
  </picture>

<p>Lookit the cuuute kitty awwww</p>
Lookit the cuuute kitty awwww
Lookit the cuuute kitty awwww
image-set()

CSS4 function
background-image:
  image-set(
    "cat.png" 1x,
    "cat-HD.png" 2x,
    "cat-print.png" 600dpi );
background-image: image-set(
  "cat.png" 1x,
  "cat-HD.png" 2x,
  "cat-print.png" 600dpi );
background-image: image-set("cat.png" 1x,
"cat-HD.png" 2x,
"cat-print.png" 600dpi);
background-image: image-set(
"cat.png" 1x,
"cat-HD.png" 2x,
"cat-print.png" 600dpi);
@media all{
  #pictureElement{
    background-image: image-set(kitten.jpg 1x, kitten-HD.jpg 2x);
  }
}

@media all and (min-width: 45em){
  #pictureElement{
    background-image: image-set(tiger.jpg 1x, tiger-HD.jpg 2x);
  }
}

@media all and (min-width: 18em){
  #pictureElement{
    background-image: image-set(cat.jpg 1x, cat.jpg 2x);
  }
}
Your favourite?
<table>
<thead>
<tr>
<th></th>
<th>Srcset</th>
<th>Image-set</th>
<th>Picture Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiple image dimensions</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
</tr>
<tr>
<td>Multiple image resolutions</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
</tr>
<tr>
<td>Reduces HTTP requests</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>
Summary
Download the source code
tinyurl.com/imageoptimiser
View the demos

tinyurl.com/LeastDemos