



Fine Tune Your Craft Performance

Feb 20, 2020 - Charlotte Craft CMS Meetup Group
Justin Holt presenter

Fine Tune Your Craft Performance

Thank You

solSPACE

Providing Food & Beverages



Providing Hosting Space

Fine Tune Your Craft Performance

Agenda

Perceived performance vs Measurable performance.

Front-end wins.

Image optimizations, tools, and techniques.

Caching.

Throw money at it!

Serverside stuff.

The Benefits.

How to measure

Fine Tune Your Craft Performance

Perceived performance.

- Responsiveness to interactions.
 - First-mile optimizations.
- If I have to wait, be sure to tell me.
(loading indicators, interaction indicators)

Fine Tune Your Craft Performance

Measurable performance.

- Time to Interactive (TTI)
 - Speed Index
- First Contentful Paint (FCP)
 - First CPU Idle
- First Meaningful Paint (FMP)
 - Estimated Input Latency
- Craft Time to First Byte (TTFB)

Fine Tune Your Craft Performance

Front-end Wins

- prefetch, preconnect, defer

```
{# Webfonts stylesheets #}  
<link rel="preload" type="text/css" href="https://cloud.typography.com/7062236/6503992/css/fonts.css" as="style" onload="this.onload=null; this.rel='stylesheet';">  
<link rel="preconnect" rel="dns-prefetch" href="https://fonts.googleapis.com/css?family=Noto+Serif" as="style" onload="this.onload=null; this.rel='stylesheet';">
```

```
{# Main stylesheet, loading asynchronously with no js fallback #}  
<link rel="preload" type="text/css" href="/styles/index.css?build=1562962714535" as="style" onload="this.onload=null; this.rel='stylesheet';">  
<noscript>  
| <link rel="stylesheet" type="text/css" href="/styles/index.css?build=1562962714535">  
</noscript>
```

```
{# Main scripts #}  
<script type="text/javascript" src="/scripts/index.js?build=1562962714535" defer></script>
```

Fine Tune Your Craft Performance

Front-end Wins - critical CSS



Style Inliner

A Twig tag for inlining styles in a template.

Free



Critical CSS

Generate Critical CSS without the need for SSH access.

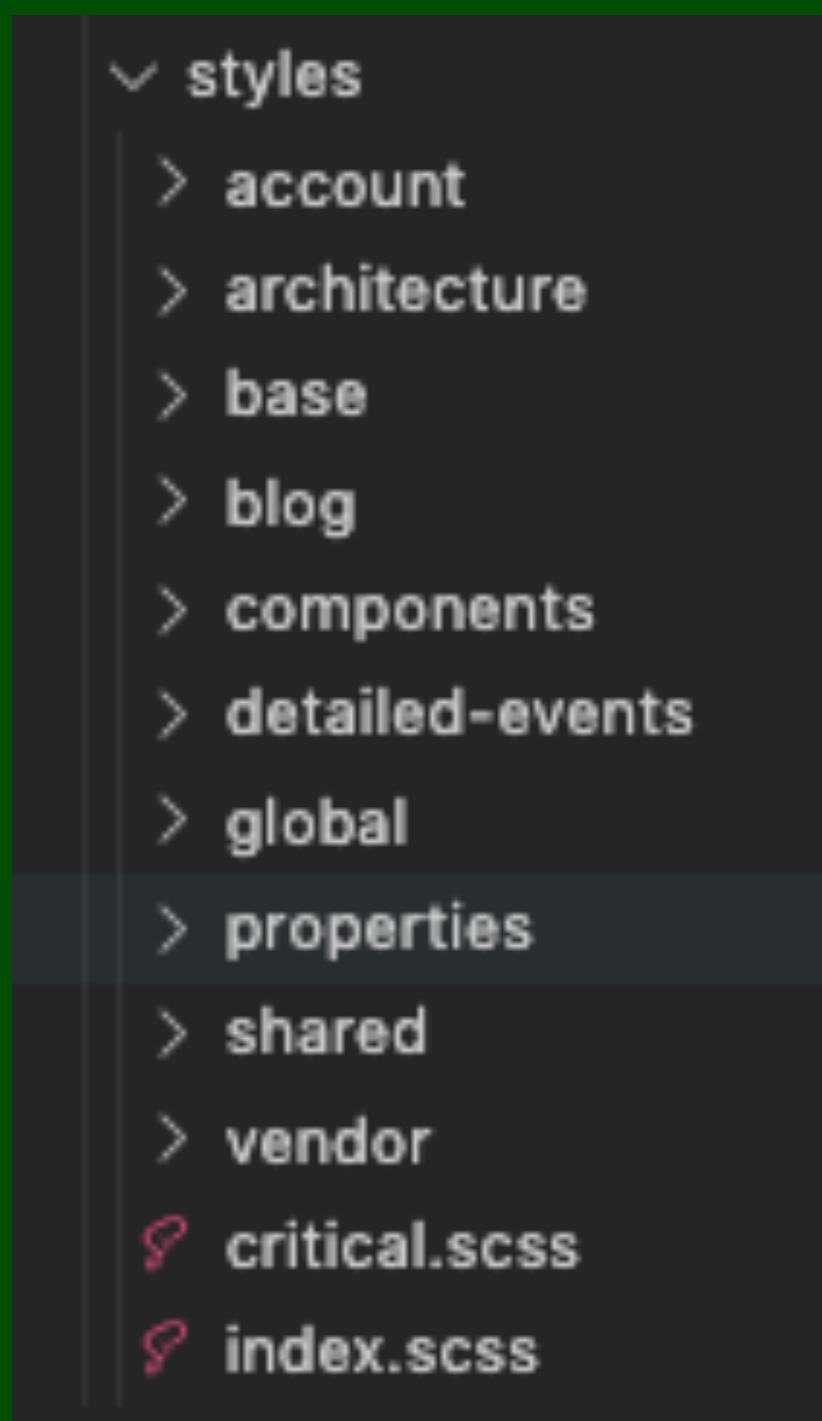
Free

Inline “above the fold” styles in head.

```
{# First mile styles inline #}  
{% criticalcss 'critical' %}
```

Fine Tune Your Craft Performance

Front-end Wins - critical CSS



```
critical.scss x
src > styles > critical.scss
1  /**
2  * styles/index.scss
3  * Entry point for included styles
4  */
5
6
7  /**
8  * Order is important in the base imports
9  */
10
11 @import "base/reset";
12 @import "base/variables";
13 @import "base/fonts";
14 @import "base/typography";
15 @import "base/mixins";
16 @import "base/animations";
17 @import "base/elements";
18 @import "base/breakpoints";
19
20 // Global
21 @import "global/load-in";
22 // Base components - place before components to allow for style overrides
23 @import "architecture/layout";
24 @import "architecture/header";
25
26 // Components
27 @import "components/event-page-subnav";
28 @import "components/form";
29 @import "components/form-modal";
30 @import "components/hero-detailed";
31 @import "components/hero-single-image";
32 @import "components/hero-slideshow";
33 @import "components/page-content";
34 @import "components/page-introduction";
35 // Properties
36 @import "properties/detail";
37 // Shared
38 @import "shared/button";
39 @import "shared/video-background";
40 @import "shared/video-hero";
41 @import "shared/video-player";
42 // Events
43 @import "detailed-events/global";
```

```
index.scss x
src > styles > index.scss
5
6
7  /**
8  * Order is important in the base imports
9  */
10
11 @import "base/reset";
12 @import "base/variables";
13 @import "base/fonts";
14 @import "base/typography";
15 @import "base/mixins";
16 @import "base/animations";
17 @import "base/elements";
18 @import "base/breakpoints";
19
20 // Global
21
22 // Base components - place before components to allow for style overrides
23 @import "architecture/footer";
24 // Components
25 @import "components/background-card";
26 @import "components/block-form";
27 @import "components/card-collection";
28 @import "components/event-modal";
29 @import "components/image-card";
30 @import "components/image-gallery";
31 @import "components/inline-form";
32 @import "components/input";
33 @import "components/instagram";
34 @import "components/issue-collection";
35 @import "components/leadership-modal";
36 @import "components/newsletter-modal";
37 @import "components/property-card";
38 @import "components/property-form";
39 @import "components/property-search";
40 @import "components/page-subnav";
41 @import "components/sponsor-scroller";
42 @import "components/tabs";
43 @import "components/talentTabs";
44 @import "components/tabsText";
45 // Shared
46 @import "shared/video-modal";
47 // Singles
```


Fine Tune Your Craft Performance

Front-end Wins

- Enforce content editing limits

```
'maxUploadFileSize' => '100000000000000000000000',
```

Don't do this!

- minify everything

(even markup with craft minify plugin)



Minify
nystudio107

Minify adds several block tags for minifying HTML, CSS, and JS inline in your templates. Minify does not minify external CSS or JS files. Use grunt or gulp task runners to set up a workflow that minimizes these as part of your build process.

You can nest any number of the various `{% minify %}` tags as you wish.

Fine Tune Your Craft Performance

Image optimizations, tools, and techniques.

"Never let your client upload an image to the website."

Bill Clinton

- Start at the editor level
(encourage the best possible optimization at desktop level, ImageOptim application)

Fine Tune Your Craft Performance

Image optimizations, tools, and techniques.

```
config > imager.php
1 <?php
2
3 return [
4     'imagerSystemPath' => '@webroot/uploads/imager/',
5     'imagerUrl' => '/uploads/imager/',
6     'cacheRemoteFiles' => false,
7     'jpegQuality' => 70,
8     'webpQuality' => 70,
9     'interlace' => true,
10    'allowUpscale' => false,
11    'removeMetadata' => false,
12    'useFilenamePattern' => false,
13
14    'optimizeType' => 'job',
15    'optimizers' => ['jpegtran'],
16    'optimizerConfig' => [
17        'jpegoptim' => [
18            'extensions' => ['jpg'],
19            'path' => '/usr/bin/jpegoptim',
20            'optionString' => '-s',
21        ],
22    ],
23    'suppressExceptions' => true,
24 ];
25
```

- WebP images
*add handler to .htaccess
- ImagerX (long live Imager)
- Responsive images



Imager X
SpaceCatNinja

Fine Tune Your Craft Performance

Image optimizations, tools, and techniques.

- Responsive images

Call custom utility template from your views.
Pass arguments in your Entry object.

```
{% if image is not empty %}
  {% include 'utilities/_image' with {
    component: {
      handle: blockHandle ~ '__image',
      initialImage: image.getUrl('cardBackgroundImageMobileSmall'),
      image: image,
      transforms: ['cardBackgroundImageMobileSmall', 'cardBackgroundImageMobileLarge', 'cardBackgroundImageTablet', 'cardBackgroundImageDesktop', 'cardBackgroundImageCinema'],
      breakpoints: [320, 768, 992, 1250, 1600],
      alttext: title,
      lazyload: true
    }
  } %}
{% endif %}
```

Fine Tune Your Craft Performance

Image optimizations, tools, and techniques.

- Responsive images

Receive arguments as params from view in utility.

```
{# Params #}  
{% set initialImage = component.initialImage|default('') %}  
{% set image = component.image|default('') %}  
{% set transforms = component.transforms|default([]) %}  
{% set breakpoints = component.breakpoints|default([]) %}  
{% set alt = component.alttext|default(component.image.filename) %}  
{% set lazy = component.lazyload|default(false) %}
```

Fine Tune Your Craft Performance

Image optimizations, tools, and techniques.

```
1 {# Markup #}
2 {% spaceless %}
3 <picture class="{{ ' ' ~ component.handle }}">
4   {% if breakpoints %}
5     {% set transformBreakpoints = [] %}
6     {% for breakpoint in breakpoints %}
7       {% set transformBreakpoints = transformBreakpoints|merge([ width: breakpoint, webPQuality: 70 ]) %}
8     {% endfor %}
9     {% set transformedImagesWebP = craft.imager.transformImage(image, transformBreakpoints, {
10       format: 'webp',
11       interlace: true,
12     }) %}
13
14     <source {% if lazy %}data-{{ endif %}srcset="{{ craft.imager.srcset(transformedImagesWebP) }}" type="image/webp">
15   {% else %}
16     {% set webpTransformedImage = craft.imager.transformImage(image, {format: 'webp', interlace: true,}) %}
17     <source {% if lazy %}data-{{ endif %}srcset="{{ include 'utilities/_srcset' with {
18       component: {
19         image: webpTransformedImage,
20         transforms: transforms,
21         breakpoints: breakpoints
22       }
23     }}" type="image/webp">
24   {% endif %}
25   <source {% if lazy %}data-{{ endif %}srcset="{{ include 'utilities/_srcset' with {
26     component: {
27       image: image,
28       transforms: transforms,
29       breakpoints: breakpoints
30     }
31   }}"
32   {% if image.extension == 'jpeg' or image.extension == 'jpg' %}
33     type="image/jpeg"
34   {% elseif image.extension == 'png' %}
35     type="image/png"
36   {% elseif image.extension == 'gif' %}
37     type="image/gif"
38   {% endif %}
39 >
```

Use Craft image transforms and ImagerX to generate your source elements.

ImagerX also can generate srcset attribute values based on the breakpoints.

Fine Tune Your Craft Performance

Image optimizations, tools, and techniques.

```
templates > utilities > ./_srcset.twig
1  {##
2  # Srcset component
3  # @param {[]Array} image
4  # @param {[]Array} transforms
5  # @param {[]Array} breakpoints (optional)
6  # @return {html}
7  #}
8
9  {# Params #}
10 {% set image = component.image|default('') %}
11 {% set transforms = component.transforms|default('') %}
12 {% set breakpoints = component.breakpoints|default([768, 992, 1250]) %}
13 {% set srcset = [] %}
14
15 {# Variables #}
16
17 {# Markup #}
18 {%- spaceless %}
19   {# setup #}
20   {% if image is not empty %}
21     {# grab images from transforms based on breakpoints #}
22     {% for transform in transforms %}
23       {% if breakpoints[loop.index - 1] <= image.getWidth(transform) %}
24         {% set srcset = srcset|merge([image.url(transform) ~ ' ' ~ breakpoints[loop.index - 1] ~ 'w']) %}
25       {% endif %}
26     {% endfor %}
27     {# output srcset #}
28     {% for row in srcset %}
29       {{ row ~ ',' }}
30     {% endfor %}
31   {% endif %}
32 {% endspaceless -%}
```

Abstracted srcset utility,

Can be handy if you need plain srcset support.

Fine Tune Your Craft Performance

Image optimizations, tools, and techniques.

Don't forget your fallbacks for initial image and lazy load.

```
{% if initialImage %}
  
{% else %}
  
{% endif %}
<noscript>
  
</noscript>
```


Fine Tune Your Craft Performance

Image optimizations, tools, and techniques.

- Eager loading

```
{% set features = craft.entries({
  section: 'atPropublica',
  featured: 1,
  orderBy: 'postDate desc',
  with: [
    'byline',
    ['imageThreeTwoLegacy', {
      withTransforms: ['threeTwo400w']
    }],
    ['imageThreeTwo', {
      withTransforms: ['threeTwo400w', 'threeTwo800w', 'threeTwo1200w', 'threeTwo1600w']
    }]
  ]
}).all() %}
```


Fine Tune Your Craft Performance

Image optimizations, tools, and techniques.

Some cautions

- php *max_execution_time* 120 seconds,
memory_limit 512M
- nginx *request_terminate_timeout* and
fastcgi_read_timeout
- increase cache durations *cacheDuration* and
cacheDurationRemoteFiles 1 year (31536000)
- use transforms intelligently, consider how many
you are generating per page

Fine Tune Your Craft Performance

Caching.

- twig/craft cache

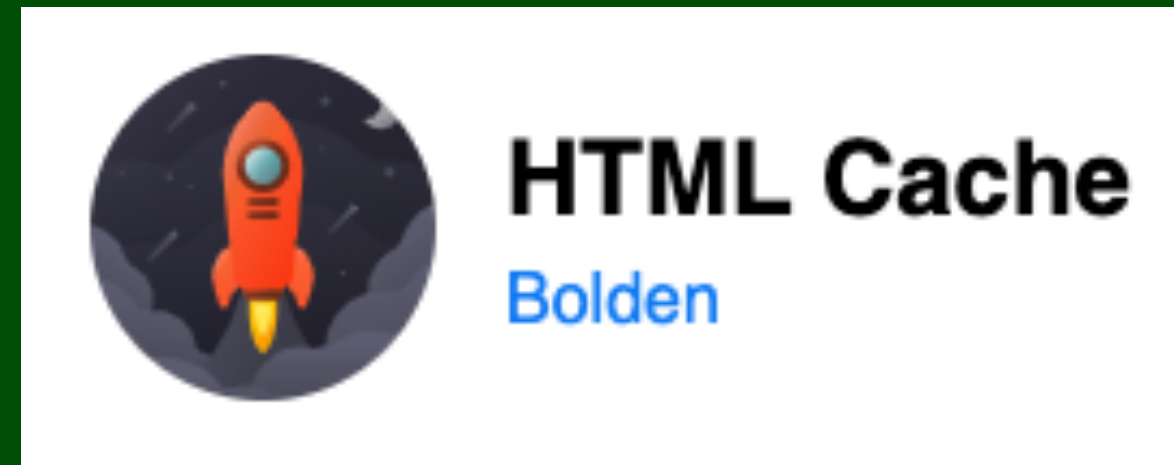
```
{% cache using key "latest-content" ~ block.id ~ craft.config.env %}
```

- Use around expensive sections (loops, matrix loops with relational blocks, Pulling data from other sites)
- DON'T USE to cache static text.

Fine Tune Your Craft Performance

Caching.

- html cache plugin

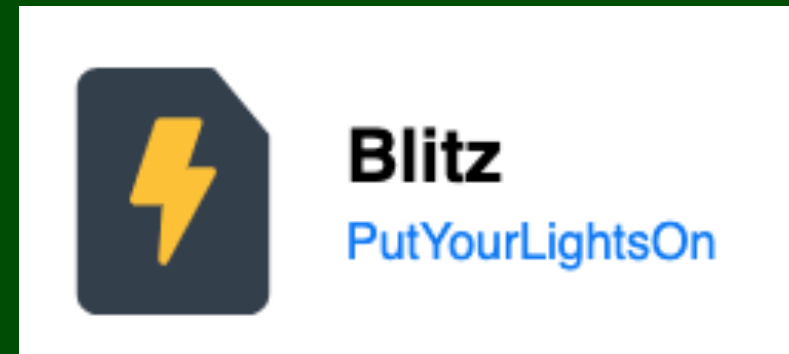


- works out of the box, no tags required.
- generates static HTML versions for any non-control panel GET request.

Fine Tune Your Craft Performance

Caching.

- blitz plugin Caching



- also caches static HTML versions of pages
 - adds cache warmers, cache storage, and reverse proxy purgers.
- \$99 paid plugin

Fine Tune Your Craft Performance

Caching.

- Cache busting



No-Cache

A Twig extension to escape caching inside cache blocks

Free



Cache Flag

Cold template caches that can be flagged and automatically

Free

Fine Tune Your Craft Performance

Throw money at it!

- CDNs (Cloudflare)
- Amazon Cloudfront
- Offload image transforms/Imgix
- Cache warm with AWS Lambda function
 - Beefy server
- More time for content optimization for editors
 - Properly scope your projects
 - Performance budgets
 - Bloat budgets

Fine Tune Your Craft Performance

Serverside stuff.

- gzip components (consider Brotli also)
 - add expires headers
- max-age or expires on static assets
 - reduce dns lookups
 - avoid redirects
 - configure ETags

Fine Tune Your Craft Performance

The benefits.

- Improved Google rankings
- YOUR WEBSITE MUST LOAD FAST ON MOBILE.
- Lower server and asset storage costs
 - Improved usability
- If you use ads, your ad quality score will go up and your ads will cost less
 - Impress your friends and family!

Fine Tune Your Craft Performance

How to measure?

- Pagespeed Insights/Lighthouse
 - GTmetrix
 - Webpagetest
 - Yslow
 - imgix page weight tools
- Pingdom website speed test
 - Flood.io

Fine Tune Your Craft Performance

Thank You

solSPACE

Providing Food & Beverages



Providing Hosting Space

Fine Tune Your Craft Performance

justinholt.com/fine-tune-performance