



Brandon Morelli [Follow](#)

Creator of @codeburstio— Frequently posting web development tutorials & articles. Follow me on Twitter too: @BrandonMorelli

Oct 29 · 12 min read

## The Front-End Checklist

An Exhaustive List of all the Elements you need to have/test Before Launching your Website To Production.



The **Front-End Checklist** is an exhaustive list of all elements you need to have / to test before launching your site / HTML page to production.

It is based on Front-End developers' years of experience, with additions coming from other open-source checklists. **David Dias** is the main author/creator and posted the guide to **GitHub** where it has taken off! The guide is also available below. Don't forget to thank [David](#) for an awesome guide!

## Table of Contents

1. Head
2. HTML
3. Webfonts
4. CSS
5. Images

6. JavaScript
7. Security
8. Performance
9. Accessibility
10. SEO

## How To Use

All items in the Front-End Checklist are required for the majority of the projects, but some elements can be omitted or are not essential (in the case of an administration web app, you may not need RSS feed for example).

Some resources possess an emoticon to help you understand which type of content / help you may find on the checklist:

- 📖 documentation or article
- 🛠️ online tool / testing tool
- 📺 media or video content

. . .

## Head

Notes: You can find a list of everything that could be found in the `<head>` of an HTML document.

### Meta tag

- Doctype: The Doctype is HTML5 and is at the top of all your HTML pages.

```
<!-- Doctype HTML5 -->
<!doctype html>
```

📖 Determining the character encoding—HTML5 W3C

*The next 3 meta tags (Charset, X-UA Compatible and Viewport) need to come first in the head.*

- Charset: The charset declared (UTF-8) is declared correctly.

```
<!-- Set character encoding for the document -->  
<meta charset="utf-8">
```

- X-UA-Compatible: The X-UA-Compatible meta tag is present.

```
<!-- Instruct Internet Explorer to use its latest rendering  
engine -->  
<meta http-equiv="x-ua-compatible" content="ie=edge">
```

### Specifying legacy document modes (Internet Explorer)

- Viewport: The viewport is declared correctly.

```
<!-- Viewport for responsive web design -->  
<meta name="viewport" content="width=device-width, initial-  
scale=1">
```

- Title: A title is used on all pages (SEO: Google calculate the pixel width of the characters used in the title, cut off between 472 and 482 pixels. Average character limit would be around 55-characters).

```
<!-- Document Title -->  
<title>Page Title less than 65 characters</title>
```

### Title—HTML—MDN

### SERP Snippet Generator

- Description: A meta description is provided, it is unique and doesn't possess more than 150 characters.

```
<!-- Meta Description -->
<meta name="description" content="Description of the page
less than 150 characters">
```

#### [Meta Description—HTML—MDN](#)

- Favicons: Each favicon has been created and displays correctly. If you have only a `favicon.ico`, put it at the root of your site. Normally you won't need to use any markup. However, it's still good practice to link to it using the example below. Today, PNG format is recommended over `.ico` format (dimensions: 32x32px).

```
<!-- Standard favicon -->
<link rel="icon" type="image/x-icon"
href="https://example.com/favicon.ico">
<!-- Recommended favicon format -->
<link rel="icon" type="image/png"
href="https://example.com/favicon.png">
```

#### [Favicon Generator](#)

#### [RealFaviconGenerator](#)

#### [Favicon Cheat Sheet](#)

#### [Favicons, Touch Icons, Tile Icons, etc. Which Do You Need?—CSS Tricks](#)

#### [PNG favicons—caniuse](#)

- Apple Touch Icon: Apple touch favicon apple-mobile-web-app-capable are present. (Create your Apple Icon file with at least 200x200px dimension to support all dimensions that you may need)

```
<!-- Apple Touch Icon -->
<link rel="apple-touch-icon" href="/custom-icon.png">
```

## Configuring Web Applications

- Windows Tiles:
- Windows tiles are present and linked.

```
<!-- Microsoft Tiles -->
<meta name="msapplication-config"
content="browserconfig.xml" />
```

Minimum required xml markup for the browserconfig.xml file is as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<browserconfig>
  <msapplication>
    <tile>
      <square70x70logo src="small.png"/>
      <square150x150logo src="medium.png"/>
      <wide310x150logo src="wide.png"/>
      <square310x310logo src="large.png"/>
    </tile>
  </msapplication>
</browserconfig>
```

## Browser configuration schema reference

- Canonical: Use `rel="canonical"` to avoid duplicate content.

```
<!-- Helps prevent duplicate content issues -->
<link rel="canonical" href="http://example.com/2017/09/a-
new-article-to-red.html">
```

## Use canonical URLs—Search Console Help—Google Support

## 5 common mistakes with rel=canonical—Google Webmaster Blog

## HTML tags

- Language attribute: The `lang` attribute of your website is specified and related to the language of the current page.

```
<html lang="en">
```

- Direction attribute: The direction of text is specified on the `html` tag (It can be used on another HTML tag).

```
<html dir="rtl">
```

 [dir—HTML—MDN](#)

- Alternate language: The language tag of your website is specified and related to the language of the current page.

```
<link rel="alternate" href="https://es.example.com/"  
hreflang="es">
```

- Conditional comments: Conditional comments are present for IE if needed.

 [About conditional comments \(Internet Explorer\)—MSDN—Microsoft](#)

- RSS feed: If your project is a blog or has articles, an RSS link was provided.
- inline critical CSS: CSS which styles content that is immediately visible during pageload (“above the fold content”) is called “critical CSS”. It is embedded before your principal CSS call and between `<style></style>` in a single line (minified).

 [Critical by Addy Osmani on Github](#) automates this

- CSS order: All CSS files are loaded before any JavaScript files in the `<head>`. (Except the case where sometimes JS files are loaded asynchronously on top of your page).


## Social meta

Facebook OG and Twitter Cards are, for any website, highly recommended. The other social media tags can be considered if you target a particular presence on those and want to ensure the display.

- Facebook Open Graph: All Facebook Open Graph (OG) are tested and no one is missing or with a false information. Images need to be at least 600 x 315 pixels, 1200 x 630 pixels recommended.

```
<meta property="og:type" content="website">
<meta property="og:url"
content="https://example.com/page.html">
<meta property="og:title" content="Content Title">
<meta property="og:image"
content="https://example.com/image.jpg">
<meta property="og:description" content="Description Here">
<meta property="og:site_name" content="Site Name">
<meta property="og:locale" content="en_US">
```


 [A Guide to Sharing for Webmasters](#)

 Test your page with the [Facebook OG testing](#)

- Twitter Card:

```
<meta name="twitter:card" content="summary">
<meta name="twitter:site" content="@site_account">
<meta name="twitter:creator" content="@individual_account">
<meta name="twitter:url"
content="https://example.com/page.html">
<meta name="twitter:title" content="Content Title">
<meta name="twitter:description" content="Content
description less than 200 characters">
<meta name="twitter:image"
content="https://example.com/image.jpg">
```

 [Getting started with cards—Twitter Developers](#)

 Test your page with the [Twitter card validator](#)

# HTML

## Best practices

- HTML5 Semantic Elements: HTML5 Semantic Elements are used appropriately (header, section, footer, main...).

### HTML Reference

- Error pages: Error 404 page and 5xx exist. Remember that the 5xx error pages need to have their CSS integrated (no external call on the current server).
- Noopener: In case you are using external links with `target="_blank"`, your link should have a `rel="noopener"` attribute to prevent tab nabbing. If you need to support older versions of Firefox, use `rel="noopener noreferrer"`.

### About rel=noopener

- Clean up comments: Unnecessary code needs to be removed before sending the page to production.

## HTML testing

- W3C compliant: All pages need to be tested with the W3C validator to identify possible issues in the HTML code.

### W3C validator

- HTML Lint: I use tools to help me analyze any issues I could have on my HTML code.

### Dirty markup

- Link checker: There are no broken links in my page, verify that you don't have any 404 error.

### W3C Link Checker

- Adblockers test: Your website shows your content correctly with adblockers enabled (You can provide a message encouraging people to disable their adblocker).



## Webfonts

- Webfont format: WOFF, WOFF2 and TTF are supported by all modern browsers.

 [WOFF—Web Open Font Format—CanIuse.](#)

 [WOFF 2.0—Web Open Font Format—CanIuse.](#)

 [TTF/OTF—TrueType and OpenType font support](#)

 [Using @font-face—CSS-Tricks](#)

- Webfont size: Webfont sizes don't exceed 2 MB (all variants included).


. . .

## CSS

Notes: Take a look at [CSS guidelines](#) and [Sass Guidelines](#) followed by most Front-End developers. If you have a doubt about CSS properties, you can visit [CSS Reference](#).

- Responsive Web Design: The website is using responsive web design.
- CSS Print: A print stylesheet is provided and is correct on each page.
- Preprocessors: Your page is using a CSS preprocessor ([Sass](#) is preferred).
- Unique ID: If IDs are used, they are unique to a page.
- Reset CSS: A CSS reset (reset, normalize or reboot) is used and up to date. *(If you are using a CSS Framework like Bootstrap or Foundation, a Normalize is already included into it.)*

 [Reset.css](#)

 [Normalize.css](#)

## Reboot

- JS prefix: All classes (or id- used in JavaScript files) begin with js- and are not styled into the CSS files.

```
<div id="js-slider" class="my-slider">
<!-- Or -->
<div id="id-used-by-cms" class="js-slider my-slider">
```

- Embedded or inline CSS: Avoid at all cost embedding CSS in `<style>` tags or using inline CSS: only use for valid reasons (e.g. background-image for slider, critical CSS).
- Vendor prefixes: CSS vendor prefixes are used and are generated accordingly with your browser support compatibility.

## Autoprefixer CSS online

### Performance

- Concatenation: CSS files are concatenated in a single file. *(Not for HTTP/2)*
- Minification: All CSS files are minified.
- Non-blocking: CSS files need to be non-blocking to prevent the DOM from taking time to load.

## loadCSS by filament group

## Example of preload CSS using loadCSS

- Unused CSS: Remove unused CSS.


## UnCSS Online


## PurifyCSS

## Chrome DevTools Coverage

## CSS testing

- Stylelint: All CSS or SCSS files are without any errors.

 [stylelint, a CSS linter](#)

 [Sass guidelines](#)

- Responsive web design: All pages were tested at the following breakpoints: 320px, 768px, 1024px (can be more / different according to your analytics).
- CSS Validator: The CSS was tested and pertinent errors were corrected.


 [CSS Validator](#)

- Desktop Browsers: All pages were tested on all current desktop browsers (Safari, Firefox, Chrome, Internet Explorer, EDGE...).
- Mobile Browsers: All pages were tested on all current mobile browsers (Native browser, Chrome, Safari...).
- OS: All pages were tested on all current OS (Windows, Android, iOS, Mac...).
- Pixel perfect: Pages are close to pixel perfect. Depending on the quality of the creatives, you may not be 100% accurate, but your page needs to be close to your template.

[Pixel Perfect—Chrome Extension](#)

- Reading direction: All pages need to be tested for LTR and RTL languages if they need to be supported.

 [Building RTL-Aware Web Apps & Websites: Part 1—Mozilla Hacks](#)

 [Building RTL-Aware Web Apps & Websites: Part 2—Mozilla Hacks](#)

. . .

## Images

Notes: For a complete understanding of image optimization, check the free ebook [Essential Image Optimization](#) from Addy Osmani.

## Best practices

- Optimization: All images are optimized to be rendered in the browser. WebP format could be used for critical pages (like Homepage).

### [Imagemin](#)

### Use [ImageOptim](#) to optimise your images for free.

- Picture/Srcset: You use picture/srcset to provide the most appropriate image for the current viewport of the user.

### [How to Build Responsive Images with srcset](#)

- Retina: You provide layout images 2x or 3x, support retina display.
- Sprite: Small images are in a sprite file (in the case of icons, they can be in an SVG sprite image).
- Width and Height: Set `width` and `height` attributes on `<img>` if the final rendered image size is known (can be omitted for CSS sizing).
- Alternative text: All `<img>` have an alternative text which describe the image visually.

### [Alt-texts: The Ultimate Guide](#)

- Lazy loading: Images are lazyloaded (A noscript fallback is always provided).

. . .

## JavaScript

### Best practices

- JavaScript Inline: You don't have any JavaScript code inline (mixed with your HTML code).
- Concatenation: JavaScript files are concatenated.

- Minification: JavaScript files are minified (you can add the `.min` suffix).

#### [Minify Resources \(HTML, CSS, and JavaScript\)](#)

- JavaScript security:

#### [Guidelines for Developing Secure Applications Utilizing JavaScript](#)

- Non-blocking: JavaScript files are loaded asynchronously using `async` or deferred using `defer` attribute.

#### [Remove Render-Blocking JavaScript](#)

- Modernizr: If you need to target some specific features you can use a custom Modernizr to add classes in your `<html>` tag.

#### [Customize your Modernizr](#)

## JavaScript testing

- ESLint: No errors are flagged by ESLint (based on your configuration or standards rules).

#### [ESLint—The pluggable linting utility for JavaScript and JSX](#)

. . .

## Security

### Scan and check your web site

[securityheaders.io](#)

[Observatory by Mozilla](#)


[ASaFaWeb—Automated Security Analyser for ASP.NET Websites](#)

### Best practices

- HTTPS: HTTPS is used on every pages and for all external content (plugins, images...).

 [Let's Encrypt—Free SSL/TLS Certificates](#)

 [Free SSL Server Test](#)

 [Strict Transport Security](#)

- HTTP Strict Transport Security (HSTS):
- The HTTP header is set to 'Strict-Transport-Security'.

 [Check HSTS preload status and eligibility](#)

 [HTTP Strict Transport Security Cheat Sheet—OWASP](#)

 [Transport Layer Protection Cheat Sheet—OWASP](#)

- Cross Site Request Forgery (CSRF): You ensure that requests made to your server-side are legitimate and originate from your website / app to prevent CSRF attacks.

 [Cross-Site Request Forgery \(CSRF\) Prevention Cheat Sheet—OWASP](#)

- Cross Site Scripting (XSS): Your page or website is free from XSS possible issues.


 [XSS \(Cross Site Scripting\) Prevention Cheat Sheet—OWASP](#)

 [DOM based XSS Prevention Cheat Sheet—OWASP](#)

- Content Type Options: Prevents Google Chrome and Internet Explorer from trying to mime-sniff the content-type of a response away from the one being declared by the server.

 [X-Content-Type-Options—Scott Helme](#)

- X-Frame-Options (XFO): Protects your visitors against clickjacking attacks.

 [X-Frame-Options—Scott Helme](#)

 [RFC7034—HTTP Header Field X-Frame-Options](#)

. . .

## Performance

### Best practices

- Weight page: The weight of each page is between 0 and 500 KB.

 [Website Page Analysis](#)

 [Size Limit: Make the Web lighter](#)

- Minified: Your HTML is minified.

 [W3C Validator](#)


- Lazy loading: Images, scripts and CSS need to be lazy loaded to improve the response time of the current page (See details in their respective sections).
- Cookie size: If you are using cookies be sure each cookie doesn't exceed 4096 bytes and your domain name doesn't have more than 20 cookies.

 [Cookie specification: RFC 6265](#)

 [Cookies](#)

 [Browser Cookie Limits](#)

- Third party components: Third party iframes or components relying on external JS (like sharing buttons) are replaced by static components when possible, thus limiting calls to external APIs and keeping your users activity private.

 [Simple sharing buttons generator](#)

### Preparing upcoming requests

### Explanation of the following techniques

- DNS resolution: DNS of third-party services that may be needed are resolved in advance during idle time using `dns-prefetch` .

```
<link rel="dns-prefetch" href="https://example.com">
```

- Preconnection: DNS lookup, TCP handshake and TLS negotiation with services that will be needed soon is done in advance during idle time using `preconnect` .

```
<link rel="preconnect" href="https://example.com">
```

- Prefetching: Resources that will be needed soon (e.g. lazy loaded images) are requested in advance during idle time using `prefetch` .

```
<link rel="prefetch" href="image.png">
```

- Preloading: Resources needed in the current page (e.g. scripts placed at the end of `<body>` ) in advance using `preload` .

```
<link rel="preload" href="app.js">
```

### Difference between prefetch and preload

## Performance testing

- Google PageSpeed: All your pages were tested (not only the homepage) and have a score of at least 90/100.

 Google PageSpeed

 Test your mobile speed with Google



. . .

## Accessibility

Notes: You can watch the playlist [A11ycasts with Rob Dodson](#) 

### Best practices

- Progressive enhancement: Major functionality like main navigation and search should work without JavaScript enabled.


 [Enable / Disable JavaScript in Chrome Developer Tools](#)

- Color contrast: Color contrast should at least pass WCAG AA (AAA for mobile).

 [Contrast ratio](#)


### Headings

- H1: All pages have an H1 which is not the title of the website.
- Headings: Headings should be used properly in the right order (H1 to H6).

 [Why headings and landmarks are so important—A11ycasts #18](#)

### Landmarks

- Role banner: `<header>` has `role="banner"` .
- Role navigation: `<nav>` has `role="navigation"` .
- Role main: `<main>` has `role="main"` .

 [Using ARIA landmarks to identify regions of a page](#)

### Semantics

- Specific HTML5 input types are used: This is especially important for mobile devices that show customized keypads and widgets for different types.

 [Mobile Input Types](#)


## Form

- Label: A label is associated with each input form element. In case a label can't be displayed, use `aria-label` instead.


 [Using the aria-label attribute—MDN](#)

## Accessibility testing

- Accessibility standards testing: Use the WAVE tool to test if your page respects the accessibility standards.

 [Wave testing](#)

- Keyboard navigation: Test your website using only your keyboard in a previsible order. All interactive elements are reachable and usable.
- Screen-reader: All pages were tested in a screen-reader (VoiceOver, ChromeVox, NVDA or Lynx).
- Focus style: If the focus is disabled, it is replaced by visible state in CSS.

 [Managing Focus—A11ycasts #22](#)

. . .

## SEO


- Google Analytics: Google Analytics is installed and correctly configured.
- Headings logic: Heading text helps to understand the content in the current page.
- sitemap.xml: A sitemap.xml exists and was submitted to Google Search Console (previously Google Webmaster Tools).
- robots.txt: The robots.txt is not blocking webpages.

 [Test your robots.txt with Google Robots Testing Tool](#)

- Structured Data: Pages using structured data are tested and are without errors. Structured data helps crawlers understand the content in the current page.


 [Introduction to Structured Data—Search—Google Developers](#)

 [Test your page with the Structured Data Testing Tool](#)

 [Complete list of vocabularies that can be used as structured data. Schema.org Full Heirarchy](#)

- Sitemap HTML: An HTML sitemap is provided and is accessible via a link in the footer of your website.

 [Sitemap guidelines—Google Support](#)

 [Sitemap generator](#)

. . .

## Author

[David Dias](#)

## License

Public Domain

## Closing Notes:

Thanks for reading! If you're ready to finally learn Web Development, check out [\*\*The Ultimate Guide to Learning Full Stack Web Development in 6 months, for \\$30.\*\*](#) If you're just looking to improve your JavaScript Skills, check out: [\*\*A Beginners Guide to Advanced JavaScript\*\*](#)

I publish 4 articles on web development each week. Please consider [\*\*entering your email here\*\*](#) if you'd like to be added to my once-weekly email list, or follow me on [Twitter](#).

If this post was helpful, please click the clap 🖐️ button below a few times to show your support! ↓ ↓





