

[HTML5 Boilerplate homepage](#) | [Documentation table of contents](#)

Usage

The most basic usage of HTML5 Boilerplate is to create a static site or simple app. Once you've downloaded or cloned the project, that process looks something like this:

1. Set up the basic structure of the site.
2. Add some content, style, and functionality.
3. Run your site locally to see how it looks.
4. Deploy your site.

Cool, right? *It is.* That said, the smart defaults, baseline elements, default attribute values and various other utilities that HTML5 Boilerplate offers can serve as the foundation for whatever you're interested in building.

Even the basic use-case of a simple static site can be enhanced by manipulating the code through an automated build process. Moving up in complexity HTML5 Boilerplate can be integrated with whatever front-end framework, CMS or e-commerce platform you're working with. Mix-and-match to your heart's content. Use what you need (toss it in a blender if you need to) and discard the rest. HTML5 Boilerplate is a starting point, not a destination.

Basic structure

A basic HTML5 Boilerplate site initially looks something like this:

```
.
├── css
│   ├── normalize.css
│   └── style.css
├── doc
├── img
├── js
│   ├── app.js
│   └── vendor
│       └── modernizr.min.js
├── .editorconfig
├── .htaccess
├── 404.html
├── browserconfig.xml
├── favicon.ico
├── humans.txt
├── icon.png
├── icon.svg
├── index.html
├── package.json
├── robots.txt
├── site.webmanifest
├── tile.png
├── tile-wide.png
└── webpack.config.js
```

What follows is a general overview of each major part and how to use them.

css

This directory should contain all your project's CSS files. It includes some initial CSS to help get you started from a solid foundation. [About the CSS](#).

doc

This directory contains all the HTML5 Boilerplate documentation. You can use it as the location and basis for your own project's documentation.

js

This directory should contain all your project's JS files. Libraries, plugins, and custom code can all be included here. It includes some initial JS to help get you started. [About the JavaScript](#).

.htaccess

The default web server configs are for Apache. For more information, please refer to the [Apache Server Configs repository](#).

Host your site on a server other than Apache? You're likely to find the corresponding server configs project listed in our [Server Configs](#) repository.

404.html

A helpful custom 404 to get you started.

browserconfig.xml

This file contains all settings regarding custom tiles for IE11 and Edge.

For more info on this topic, please refer to [Microsoft's Docs](#).

.editorconfig

The `.editorconfig` file is provided in order to encourage and help you and your team to maintain consistent coding styles between different editors and IDEs. [Read more about the .editorconfig file](#).

index.html

This is the default HTML skeleton that should form the basis of all pages on your site. If you are using a server-side templating framework, then you will need to integrate this starting HTML with your setup.

Make sure that you update the URLs for the referenced CSS and JavaScript if you modify the directory structure at all.

humans.txt

Edit this file to include the team that worked on your site/app, and the technology powering it.

package.json

Edit this file to describe your application, add dependencies, scripts and other properties related to node based development and the npm registry

robots.txt

Edit this file to include any pages you need hidden from search engines.

Icons

Replace the default `favicon.ico` , `tile.png` , `tile-wide.png` and Apple Touch Icon with your own.

If you want to use different Apple Touch Icons for different resolutions please refer to the [according documentation](#).