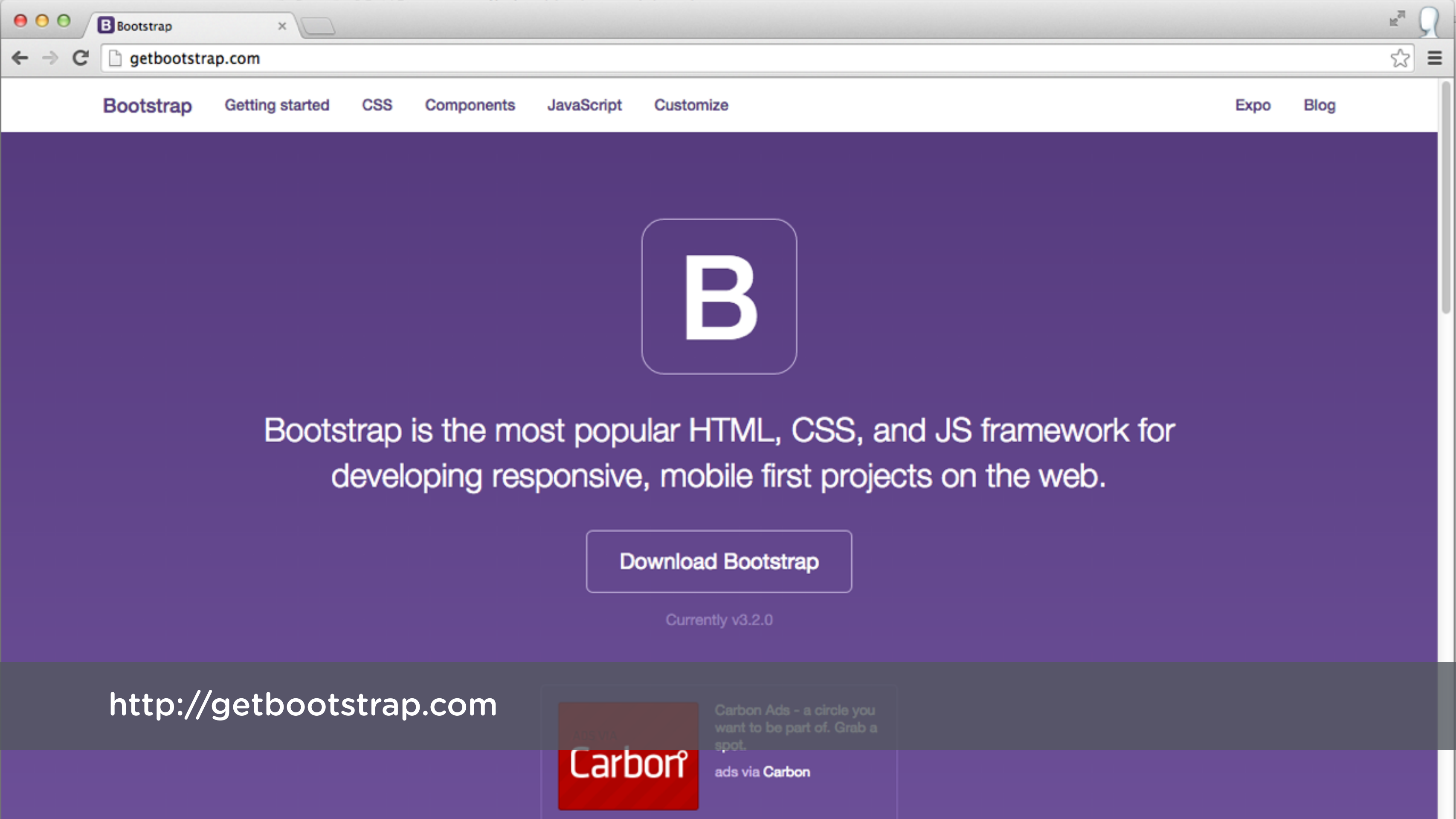


# Host Your Portfolio on GitHub...for Free!

# What are GitHub Pages?



Bootstrap

Getting started

CSS

Components

JavaScript

Customize

Expo

Blog

B

Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web.

Download Bootstrap

Currently v3.2.0

<http://getbootstrap.com>

ADS VIA  
Carbon

Carbon Ads - a circle you want to be part of. Grab a spot.

ads via Carbon



# NODESCHOOL

Open source workshops built with Node that run in your terminal. Do them on your own or at a workshop nearby.

[Events](#)[Chapters](#)[About](#)[Build](#)[Host](#)

**Get started with the Learn You Node workshopper:**

<http://nodeschool.io>

```
npm install -g learnyounode
```

[HOME](#)[ARTICLES](#)[PRESENTATIONS](#)[RESOURCES](#)[SANDBOX](#)

# WebComponents.org

a place to discuss and evolve web component best-practices

## WHAT?

WebComponents.org is where pioneers and community-members of the Web Components ecosystem (like *Polymer*, *X-tags*, and other interested parties) document web components best practices so that others can follow the same path instead of needlessly striking out on their own.

## SPECS



### WEB COMPONENTS

This document is a non-normative reference, which provides an overview of Web Components. It summarizes the normative information in the respective specifications in easy-to-digest prose with illustrations.



### CUSTOM ELEMENTS

This specification describes the method for enabling the author to define and use new types of DOM elements in a document.



### HTML IMPORTS

## ARTICLES

DEV.BYTES

## Web Components Template

### INTRODUCTION TO THE TEMPLATE ELEMENTS

Templates allow teams to divide their work, allowing designers to focus on views written in HTML and CSS, while other engineers focus on logic and back end integration.

In this article, you'll learn how to use a new HTML element that is already available on Chrome, Opera, Firefox and Safari.

<http://webcomponents.org/>

## BROWSER SUPPORT

CHROME

OPERA

FIREFOX

SAFARI

IE



# Limitations

No server-side: no ruby, php, python

Static Websites only: HTML, CSS, JavaScript

Can use Jekyll: <http://jekyllrb.com>

# Project Pages



# Parsing and Displaying a Markdown File with CommonMark

## What's This Do?

Renders a markdown file as HTML and displays it. JavaScript loads a markdown file, parses it and displays it as HTML inside the index.html file. The index.html file includes 2 Javascript files:

- The first -- `js/stdmd.js` -- loads the [CommonMark](#) library which can convert CommonMark (standardized markdown) to HTML.
- The second -- `render.js` -- loads a specified markdown file (default is `README.md`) via XHR then parses it and renders it with the CommonMark JS library.

## Why?

I wanted a way to display markdown files as HTML on any website -- but in particular, I wanted to be able to use Github pages to render the README.md file for a repo. Mainly so I could use a repo as a way to create installation guides and other how-to's in a single markdown file.

<http://sawmac.github.io/display-markdown>

I also wanted a way that was backend agnostic so it could all be handled via JavaScript without any server requirements.

## How to use



# Parsing and Displaying a Markdown File with CommonMark

## What's This Do?

Renders a markdown file as HTML and displays it. JavaScript loads a markdown file, parses it and displays it as HTML inside the index.html file. The index.html file includes 2 Javascript files:

- The first -- `js/stdmd.js` -- loads the [CommonMark](#) library which can convert CommonMark (standardized markdown) to HTML.
- The second -- `render.js` -- loads a specified markdown file (default is `README.md`) via XHR then parses it and renders it with the CommonMark JS library.

## Why?

I wanted a way to display markdown files as HTML on any website -- but in particular, I wanted to be able to use Github pages to render the README.md file for a repo. Mainly so I could use a repo as a way to create installation guides and other how-to's in a single markdown file.

<http://sawmac.github.io/display-markdown>

I also wanted a way that was backend agnostic so it could all be handled via JavaScript without any server requirements.

## How to use

username

repository name

<http://sawmac.github.io/display-markdown>

github pages domain

# User Pages

# Sawmac.github.io

Project maintained by [sawmac](#)

Hosted on GitHub Pages — Theme by [mattgraham](#)

## Welcome to GitHub Pages.

This automatic page generator is the easiest way to create beautiful pages for all of your projects. Author your page content here using GitHub Flavored Markdown, select a template crafted by a designer, and publish. After your page is generated, you can check out the new branch:

```
$ cd your_repo_root/repo_name
$ git fetch origin
$ git checkout gh-pages
```

If you're using the GitHub for Mac, simply sync your repository and you'll see the new branch.

## Designer Templates

We've crafted some handsome templates for you to use. Go ahead and continue to layouts to browse through them. You can easily go back to edit your page before publishing. After publishing your page, you can revisit the page generator and switch to another theme. Your Page content will be preserved if it remained markdown format.

<http://sawmac.github.io/>

## Rather Drive Stick?

If you prefer to not use the automatic generator, push a branch named `gh-pages` to your repository to create a page manually. In addition to supporting regular HTML content, GitHub Pages support Jekyll, a simple, blog aware static site generator written by our own Tom Preston-Werner. Jekyll makes it easy to create site-wide headers and footers without having to copy them across every page. It also offers intelligent blog support and other advanced templating



# Sawmac.github.io

Project maintained by [sawmac](#)

Hosted on GitHub Pages — Theme by [mattgraham](#)

## Welcome to GitHub Pages.

This automatic page generator is the easiest way to create beautiful pages for all of your projects. Author your page content here using GitHub Flavored Markdown, select a template crafted by a designer, and publish. After your page is generated, you can check out the new branch:

```
$ cd your_repo_root/repo_name
$ git fetch origin
$ git checkout gh-pages
```

If you're using the GitHub for Mac, simply sync your repository and you'll see the new branch.

## Designer Templates

We've crafted some handsome templates for you to use. Go ahead and continue to layouts to browse through them. You can easily go back to edit your page before publishing. After publishing your page, you can revisit the page generator and switch to another theme. Your Page content will be preserved if it remained markdown format.

<http://sawmac.github.io/>

## Rather Drive Stick?

If you prefer to not use the automatic generator, push a branch named `gh-pages` to your repository to create a page manually. In addition to supporting regular HTML content, GitHub Pages support Jekyll, a simple, blog aware static site generator written by our own Tom Preston-Werner. Jekyll makes it easy to create site-wide headers and footers without having to copy them across every page. It also offers intelligent blog support and other advanced templating

username

<http://sawmac.github.io/>

github pages domain

## project pages

All files in a **gh-pages** branch of a repo will appear on github.io



# Project Pages How-To

Create a **gh-pages** branch

**Make any changes** that you want to appear for the github pages site

index.html

css

js

**Push gh-pages** branch to GitHub

# Create gh-pages Branch

```
git checkout -b gh-pages
```

# Push gh-pages branch

```
git push origin gh-pages
```

...wait 10 minutes...

# Tips

You can create a new empty branch

```
git checkout --orphan gh-pages
```

Delete a remote branch

```
git push origin :gh-pages
```

## user pages

A repo named **username.github.io**  
will appear online as `username.github.io`

# User Pages How-To

Create a new repo: **username.github.io**

Create the site: index.html, etc.

Create local git repo

Add remote repo

Push to GitHub



Create new repo: **pdx-fl.github.io**

Create the site

# Create local repo, stage and commit

```
git init
```

```
git add .
```

```
git commit -m "Initial commit"
```

# Add remote repository (connect to GitHub)

```
git remote add origin git@github.com:username/username.github.io.git
```

# Push to GitHub

```
git push origin master
```

...wait 10 minutes...

# Add a CNAME Record

Create a CNAME alias with your DNS provider:

`www`      `pdx-f11.github.io`

Add a CNAME file to your repo

`www.pdx-robotics.club`

<https://help.github.com/articles/setting-up-a-custom-domain-with-github-pages/>



Use GitHub's automatic  
page generation feature

# Automatic Page Generator

Click Settings on the Repo

Click "Automatic Page Generator"

Fill out the form

Pick a template



This repository Search

Explore Gist Blog Help

pdx-fll



pdx-fll / 2014

# New project site

Create a new GitHub Pages site for your project.

## Project name

2014

## Tagline

## Body

h1 h2 h3



B

/



Load README.md

Markdown icon Parsed as Markdown

### Welcome to GitHub Pages.

This automatic page generator is the easiest way to create beautiful pages for all of your projects. Author your page content here using GitHub Flavored Markdown, select a template crafted by a designer, and publish. After your page is generated, you can check out the new branch:

...



<https://github.com/sawmac/github-pages>