CSS Frameworks and You:

A guide to integrating responsive frameworks into your React projects

Mario Jimenez

About me

- Been attending meetup groups for years. Decide to contribute
- Former Software Engineer at Socially Elite Pro
- Volunteer Software Engineer for Austin Public Library. Formally volunteered for Open Austin
- Sometimes build SaaS and online businesses - DevOpsJobs.app | APIplaza.co | PetWarez Inc.
- Tech blogger: https://medium.com/@7adam7e
- Sometimes create music. Sometimes do photography

What the presentation is about:

- CSS frameworks. What are they all about?
- How you can use them in most of your projects?
- Multiple CSS frameworks at use by different websites and companies
- Getting you into using a framework via a step by step sample project

What I hope that you get from this presentation

- Improve your Front-End / Web Development and Web Designing
- Enhance your communication with your UI/UX Designer, Product Managers, etc, about webpage elements
- Attract more users, traffic, or sales to your service using some of the frameworks
- Improve the UI/UX design and visual interactivity of your website, app, project, SaaS, etc
- Stop spending a lot of time center aligning website elements

What are CSS frameworks?

- Ready to use code rules and parameters used for Front-End Dev Work
- Comes with its own styling, layouts, and font styling for each text element
- Lets you easily create responsive grids, forms, buttons, and other elements for a website
- Provides consistent styling for your Front-End Dev work
- Great for adding responsive design, grids and mobile friendly CSS parameters to your project

Where you can implement new frameworks:

- Small web applications
- Legacy apps
- Progressive Web Applications
- SaaS
- Sales pages / landing pages
- Personal websites
- Codebases with separate CSS libraries referenced

Pros of using frameworks

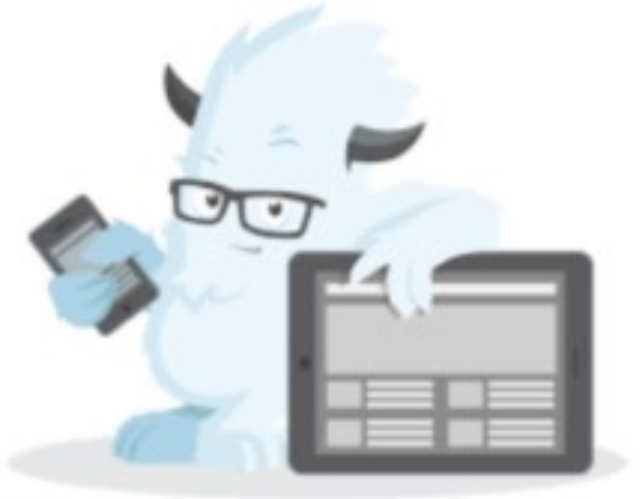
- Extremely small libraries. Some take only **2 kilobytes** of data to install (milligram.io)
- Cut down Dev time
- Ease up testing and user experiences on multiple platforms (smartphones, desktops, tablets, etc)
- Easy to update and change each framework
- They work on any type of JS/React/JS framework project. Dev stack agnostic
- Free to use, try and install the majority of frameworks (FOSS)
- Framework documentation and guides are always available

Cons of CSS Frameworks

- CSS that’s already installed in projects loses priority
- Sometimes rigid for Front-End Development. The CSS rules that the framework become the main rules
- It can turn into a mess of code if you don’t take good care of your codebase and organize everything

Some example frameworks

There's many options to choose from:



A few companies and startups that use CSS frameworks:

- **Bootstrap:** Mastercard, Spotify, LinkedIn, Twitter, Duolingo, Udemy, Robinhood
- **Tailwind:** Shopify, Medium
- **Bulma:** Fujitsu, Infosys, University of California in Berkley
- **Skeleton:** The Range, WikiTree
- **Foundation:** Barclays Bank, Pixar, Mini Cooper
- **Milligram**: Node.js foundation, Airform

Frameworks are all available on npm.js

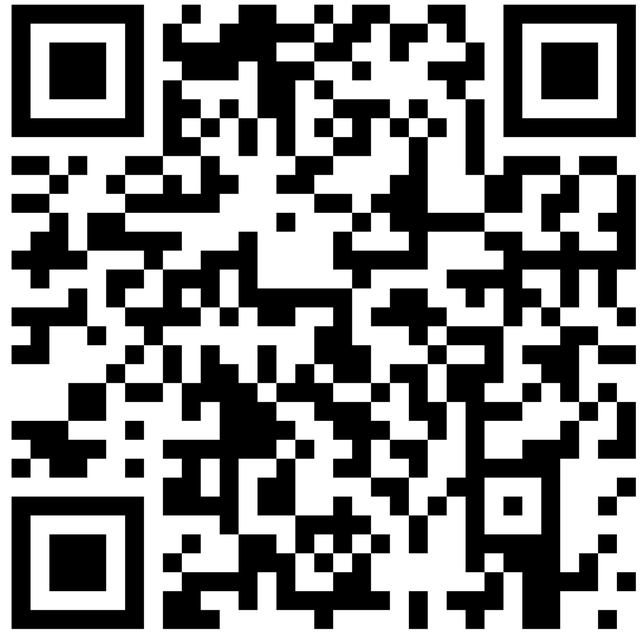
Free and easy to install on any React project

Some example websites and components from some of the frameworks

- https://bulma.io/expo/
- https://icons.getbootstrap.com/
- http://getskeleton.com/examples/landing/
- https://mui.com/material-ui/getting-started/templates/
- https://tailwindui.com/components?utm_medium=navigation&utm_source=tailwindcss

Demo project

- URL: https://github.com/tjdev7/reactatx-css-frameworks-samples
- GitHub Commands:
- https://github.com/tjdev7/reactatx-css-frameworks-samples.git
- Git clone git@github.com:tjdev7/reactatx-css-frameworks-samples.git
- gh repo clone tjdev7/reactatx-css-frameworks-samples
- Shortened URL: https://bit.ly/40Li9rz



My small advice

- Read the documentation. Follow the walkthrough guides before doing anything
- Don’t rush development. You will get a messed up codebase if you do
- Use a separate Git branch / container / directory to test things out
- Experiment, then test everything on all test mediums before pushing to main/master. Take a few minutes off before making the push with fresh eyes
- If a mistake ever happens, just revert the recent push

Conclusion

- I created this presentation with the hope that it helps you, the attendee improve your Dev cycle. And to help people in your company, group, stealth company, etc.
- I really do hope my presentation helped you out in any way
- Reach out to me if you need help. I can answer whenever I’m available via email or LinkedIn page.

- Hope you liked it
- Hope it helped you out personally
- Hope that you make some great work with it

Reach out to me:

- Github: https://github.com/tjdev7/
- Website: https://tjdev7.co/
- Twitter/X: https://twitter.com/Tjdev7
- LinkedIn: https://www.linkedin.com/in/mariojimenez1/
- Medium: https://medium.com/@7adam7e

Additional resources

- https://mui.com/
- https://getbootstrap.com/
- https://tailwindcss.com/
- https://get.foundation/
- https://bulma.io/
- http://getskeleton.com/
- https://getuikit.com/
- https://stackshare.io/tailwind-css
- https://milligram.io/typography.html
- https://www.beercss.com/
- https://jenil.github.io/chota/
- https://stackoverflow.design/
- https://semantic-ui.com/
- https://pure-css.github.io/

Npm links

- https://www.npmjs.com/package/bootstrap
- https://www.npmjs.com/package/tailwindcss
- https://www.npmjs.com/package/uikit
- https://www.npmjs.com/package/bulma
- https://www.npmjs.com/package/@mui/material