Generating Your Own Front-End Project Template

Do we even need a template or tools?

- For HTML/CSS projects we don't really need any tools at all.
- We could just have a `.html` and a `.css` file in a directory and open the file itself in our browser.
- Tools give us:
  - Code reloading
  - Optimization
  - Ability to easily add third party code (CSS frameworks, CSS resets, etc.)
  - Automation steps

app-app is dead, long live app-app

For many cohorts, SDG used app–app a JavaScript based tool that created template-based projects for students.

Benefits:

- Curated environments with just the right tools
- Automatically created a `git` project and sent code to `github`
- Built in steps for deploying

Downsides:

- Not an "industry wide tool"
- SDG has to upkeep the app as tools and libraries change versions

Our new approach

Use industry "standard" front-end code bundlers.

Benefits:

- Tools are kept up to date
- Faster and easier to install
- Industry "standard"

Downsides:

- New tools created every day (not literally, but very close)
- Requires more effort to get setup
- Less automation for steps like creating git setup and github repository

Tool of choice: vite

- Fast
- Built on latest tools
- Supports all our technologies out of the box
- Easy to customize

See: vite's homepage

Getting started - SHORTCUT MODE

To skip the rest of this setup and simply *duplicate* what SDG already setup:

```
cd ~/sdg
degit suncoast-devs/html-css-ts-project-template
cd html-css-ts-project-template
git init
git add .
git commit -m "Initial commit"
hub create
git push origin HEAD
```

And you will have your own copy of SDG's project template.

Getting started

In the same directory where you create projects:

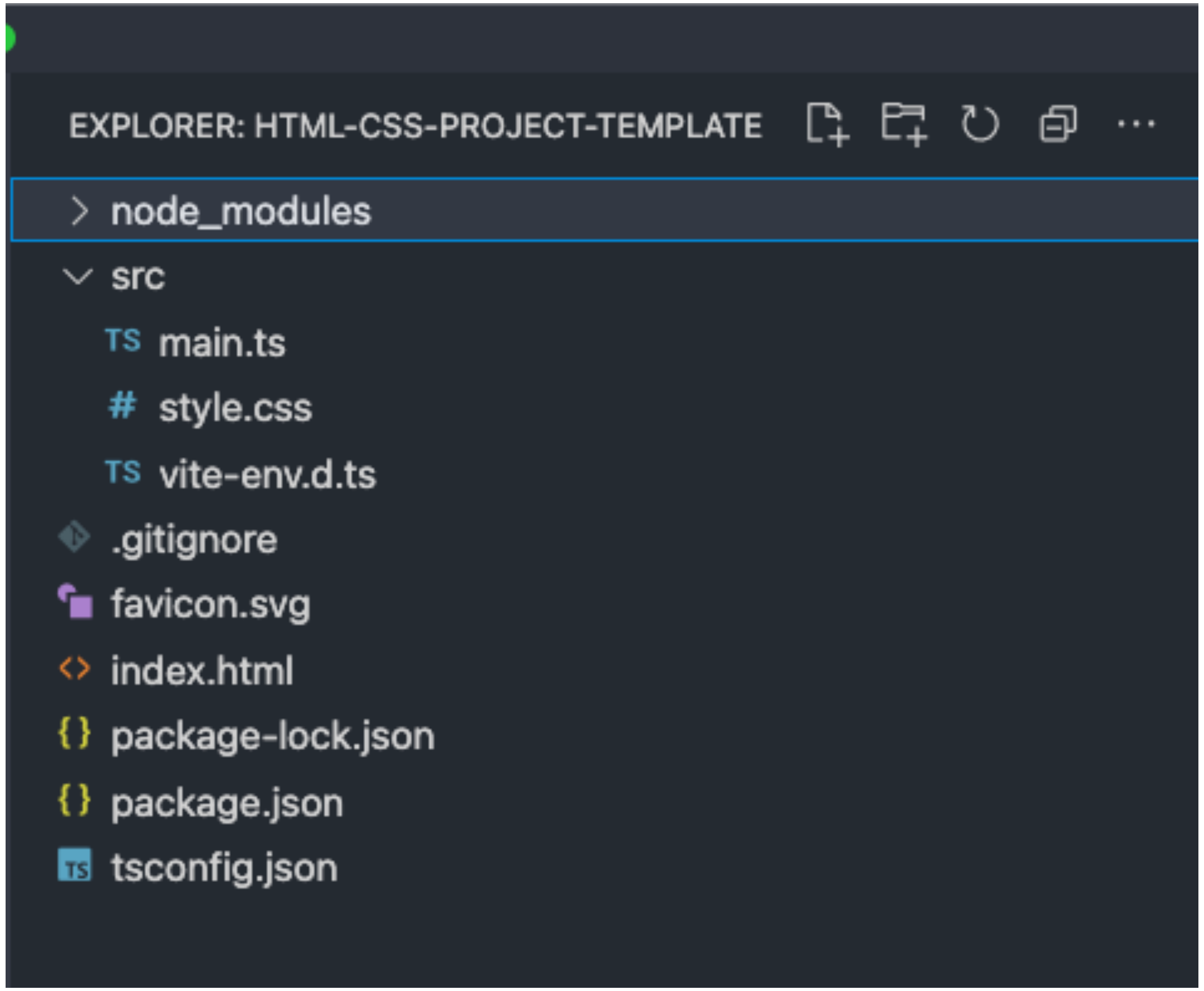
```
npm init vite -- html-css-ts-project-template
```

- Choose **vanilla** as the *framework*
- Choose **vainlla-ts** as the *variant*

Check out the template it created

```
cd html-css-ts-project-template

npm install

git init

git add .

git commit -m "Initial vite template"

code .
```

Project Structure



EXPLORER: HTML-CSS-PROJECT-TEMPLATE

```
> node_modules
∨ src
    TS main.ts
    # style.css
    TS vite-env.d.ts
  .gitignore
  favicon.svg
  <> index.html
  {} package-lock.json
  {} package.json
  tsconfig.json
```

Review files

File	Purpose
`index.html`	Main HTML content
`favicon.svg`	"favicon" graphic uses as the website icon
`package.json`	Project definition file where we list configuration and code dependencies
`package-lock.json`	Records the **specific** version of dependencies this project uses
`tsconfig.json`	Configuration file for TypeScript (see later)
`.gitignore`	Configuration for `git` to ignore files that do not belong in our project history
`src/main.ts`	Place to put our TypeScript code when we write some - And a file that loads our CSS content
`src/style.css`	Our CSS
`src/vite-env.d.ts`	Configuration for Vite itself

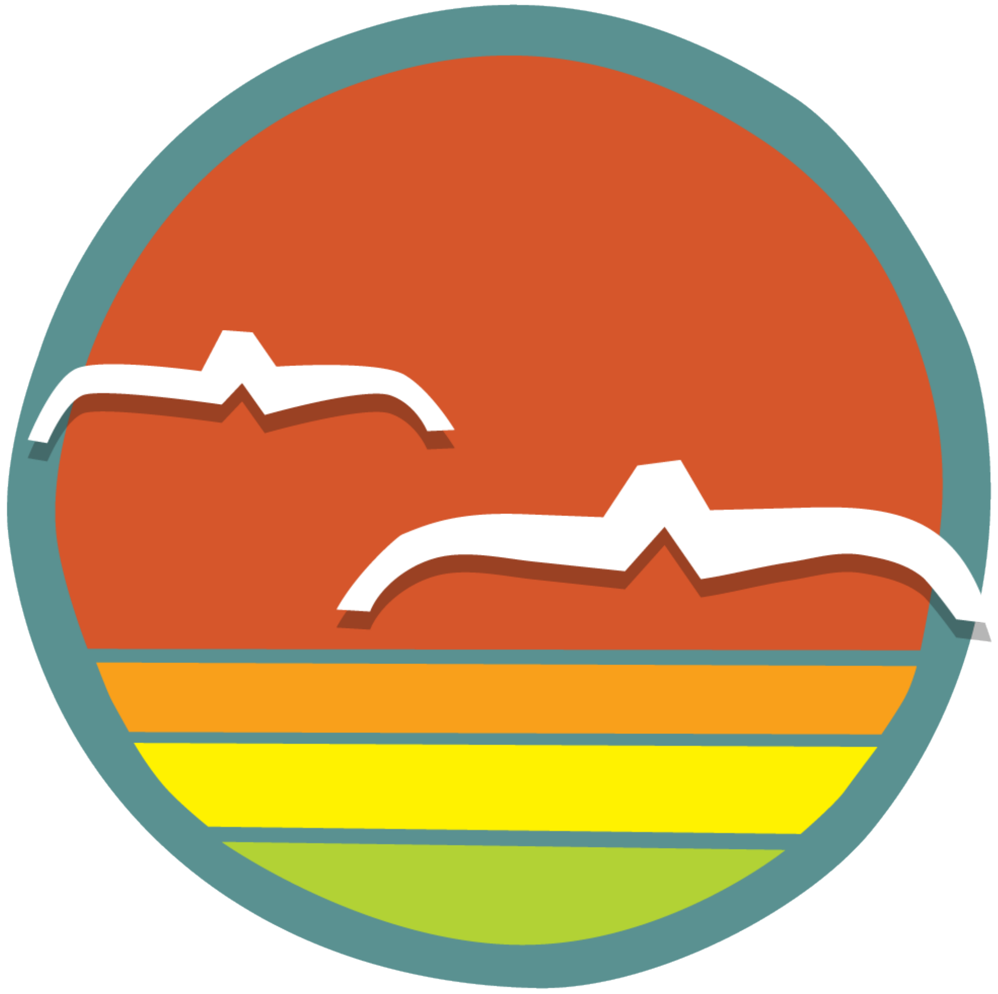
Customize!

`index.html`

- Change the default `<title>`
- Remove `<div id="app"></div>`
- Add any default HTML you think you'll want in *EVERY* template to start. Perhaps a `<p>Hello, World</p>` so something is on screen.

favicon.svg

Suggest SDG's badge graphic available here



src/main.ts

Remove all the lines **except** for the following:

```
import './style.css'
```

src/style.css

Use the following starter CSS:

```css
:root {
  font: 16px / 1 sans-serif;
}

html {
  height: 100%;
}

body {
  margin: 0;
  min-height: 100%;
}
```

`package.json`

Individual changes (example file on the next slide)

Update the `scripts` section to include:

```
"start": "vite",
"predeploy": "npm run build",
"deploy": "netlify deploy --prod --dir=dist",
```

Add this section:

```
"prettier": {
  "trailingComma": "es5",
  "tabWidth": 2,
  "semi": false,
  "singleQuote": true,
  "useTabs": false
}
```

package.json

```json
{
  "version": "0.0.0",
  "scripts": {
    "start": "vite",
    "predeploy": "npm run build",
    "deploy": "netlify deploy --prod --dir=dist",
    "dev": "vite",
    "build": "tsc && vite build",
    "serve": "vite preview"
  },
  "devDependencies": {
    "typescript": "^4.3.2",
    "vite": "^2.4.2"
  },
  "prettier": {
    "trailingComma": "es5",
    "tabWidth": 2,
    "semi": false,
    "singleQuote": true,
    "useTabs": false
  }
}
```

Ensure the latest version of `vite` and `typescript`

```
npm install --save-dev vite typescript
```

Turn on TypeScript checking in our terminal

```
npm install --save-dev vite-plugin-checker
```

Create the file `vite.config.ts` with the following:

```
import { defineConfig } from 'vite'
import checker from 'vite-plugin-checker'

// https://vitejs.dev/config/
export default defineConfig({
  plugins: [checker({ typescript: true })],
})
```

Capture initial version of our customized template

```
git add .

git commit –m "Initial template customization"
```

Create a github repository for the project

```
hub create
```

```
git push origin HEAD
```

Congratulations

You have just created your own initial starter project for the `HTML` and `CSS` projects we'll be doing in this course.

You can return to this repository and make any changes you'd like to become your new default template.

Perhaps you'll develop a color palette you like to use in each project. Come back to this template and add that CSS. All **new** projects you create will have that CSS.

More templates!

When we start with `TypeScript` we will use the same template.

We'll create a *separate* template when we start with **React**

Using your template

```
cd ~/sdg
degit YOUR-GITHUB-GOES-HERE-DONT-TYPE-THIS/html-css-ts-project-template   FirstSampleProject
```

(e.g.)

```
cd ~/sdg
degit jane-doe/html-css-ts-project-template FirstSampleProject   FirstSampleProject
cd
```

This will make a new, clean, empty, NON GIT, repository

Make this a git repository

```
cd FirstSampleProject
git init
```

Make a first initial commit to capture the base template

```
git add .
git commit -m "Initial commit"
```

Turn this into a GitHub project

```
hub create
```

Push our first commit to GitHub

```
git push origin HEAD
```

All together now…

```
degit YOUR-GITHUB-GOES-HERE-DONT-TYPE-THIS/html-css-ts-project-template   FirstSampleProject
cd FirstSampleProject
npm install
git init
git add .
git commit -m "Initial commit"
hub create
git push origin HEAD
```

Running the project!

We will use `npm start` to run the project itself. This is similar to `npm watch run` from `C#`

```
npm start
```

Open the browser

The terminal will give you the URL where the application is running, typically `http://localhost:3000`

Open that URL in your browser.

Update some code

You should see the browser *automatically refresh*!