Overview

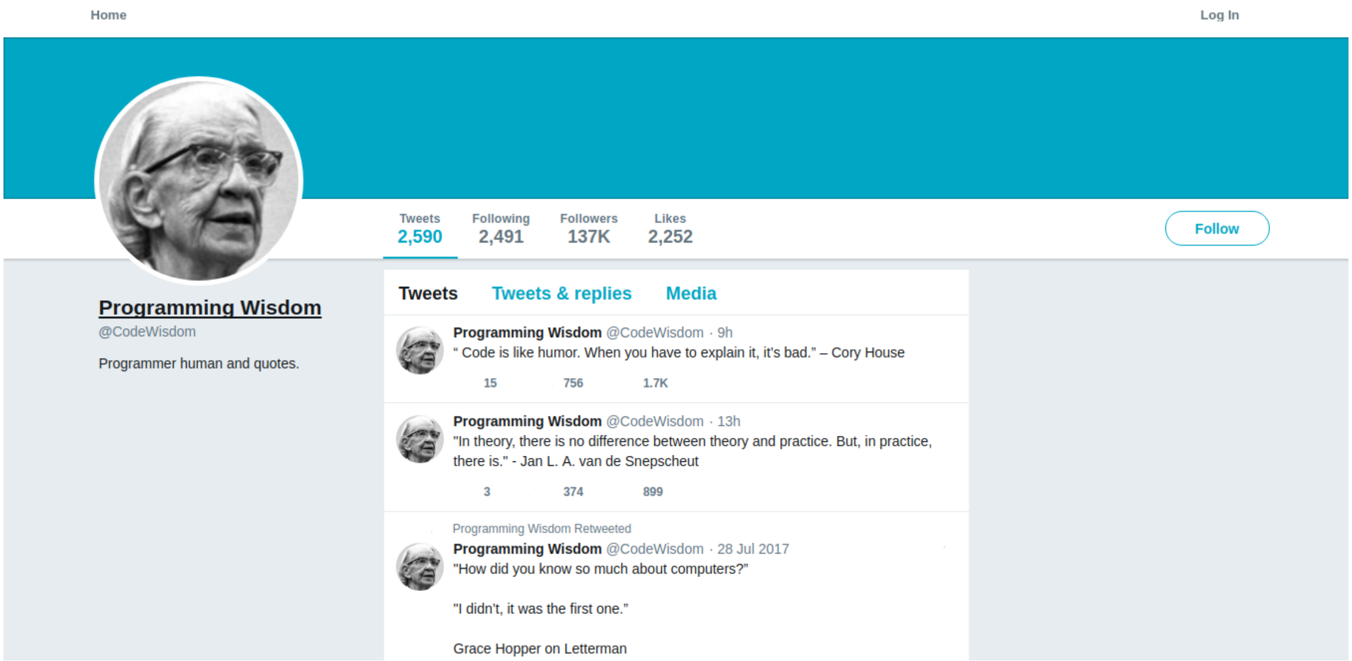
In the industry, it is a day-to-day task for front-end engineers to receive a mock-up from the design team, which consists typically of a PNG describing the shape and style of the webpage you need to build.

In Alternative Homework 1 you will be doing exactly this: Building a copy of a Twitter profile page from an image.

For more information on alternative homeworks, see "Appendix: Alternative Homework" at the end of the document.

1 Mockup

Your goal for this homework is to build the following mockup:



Programming Wisdom
@CodeWisdom

Programmer human and quotes.

Programming Wisdom @CodeWisdom · 9h
" Code is like humor. When you have to explain it, it's bad." – Cory House
15 756 1.7K

Programming Wisdom @CodeWisdom · 13h
"In theory, there is no difference between theory and practice. But, in practice, there is." - Jan L. A. van de Snepscheut
3 374 899

Programming Wisdom Retweeted
Programming Wisdom @CodeWisdom · 28 Jul 2017
"How did you know so much about computers?"

"I didn't, it was the first one."

Grace Hopper on Letterman

Download PNGs of the mockup and the Grace Hopper image used from here:
`michaelpb.github.io/sols/`

2 Requirements

- **Must be deployed to GitHub Pages**
- **Must be graphically identical to the mockup**
- Must be in a separate repo[1]

[1] Do not name your GitHub repo "homework" or something like that. Instead, give it a relevant name about its contents. This is especially important for those seeking a career change: Your GitHub profile will look less professional if all your projects seem only related to homework.

- Must use a separate CSS file to contain the style
- Must use CSS grid and/or flex-box for layout
- Must not use Bootstrap or any other pre-made CSS libraries[2]
- Must use at least one of the following concepts (preferably both):
  - BEM naming convention
  - CSS Variables

2.1 Soft requirements

- Must have clean code with a sprinkling of concise comments[3]
- Must be at least mildly responsive[4]
- Pixel-perfect (that is, literally identical down to every pixel) is a nice-to-have – in the industry, some perfectionist designers might get grumpy unless you represent their designs correct to every pixel!

3 Submission

1. Include a link to both the code view of your repo, and the deployed GitHub page
2. Include a screenshot of it working locally on your computer

3.1 Directory structure

The directory structure should look something like this:

```
- css/
    - reset.css                       (optional)
    - site.css
- img/
    -                                 (optional: any images)
- js/
    -                                 (optional: any future JS files)
- index.html
- README.md
```

4 Steps

These steps are given as a loose structure for your work.

[2] CSS resets are okay, as is using online examples to learn and build your own CSS. It's only using CSS libraries such as Bootstrap that is prohibited.

[3] Good comments are written in the present tense, imperative mood, omitting most articles, as though you were giving instructions to the computer what it is doing. For example, "Fetch currency data from API and then display" is a good comment. Also, before submission, commented-out code should be removed.

[4] That is to say, it shouldn't be stuck in one set of dimensions, but be flexible when the window is resized

4.1 Step 1: Wireframe the grid

First thing you must do is draw a grid over the mockup to figure out the grid. You should divide it into rows and columns.

4.2 Step 2: Code the grid

Using the notes you created from Step 1, and the power of CSS grids (and possibly CSS flexbox), get some `divs` with dummy content in all the right places. Worry first about broad swaths and positioning your grid, worry later about tweaking exact margins, content, and colors.

4.3 Step 3: Add in content

In all the spaces you left, add in the words and images found in the mockup.

4.4 Step 4: Tighten-up design

Examine your page closely and adjust as needed to get the colors, locations, and margins all correct to resemble the mockup.

5 Tips

- Use an "eye dropper" or color-picker tool in your image editor (such as GIMP or Photoshop) to figure out the colors used.[5]
- Use a "ruler" tool in your image editor to figure out sizes and margins[6]

6 Following CSS best practices

To get full credit for this assignment, you must follow good CSS practices.

- You must use cutting-edge CSS: use CSS grid layout and/or CSS flexbox layout somewhere in your application.
- You must research and do at least one of the following cutting-edge CSS best practices: **CSS Variables** and/or **BEM naming convention**

6.1 CSS Variables

CSS Variables let you DRY out your CSS by putting values you reuse into variables defined at the top of your CSS file, and then re-using those variables throughout other rules. It's great for keeping your color scheme separate from the rest of your file.

As an example:

[5] You can also stand-alone applications to do this, such as `gpick` for Linux

[6] Again, you can also stand-alone applications for this.

```
:root {
    --bg-color: #FEAAFE;
    --page-spacing: 50px;
}

.MainContent {
    background-color: var(--bg-color);
    margin-left: var(--page-spacing);
}

.NavBar {
    padding: var(--page-spacing);
}
```

Read more here:

developer.mozilla.org/en-US/docs/Web/CSS/Using_CSS_variables

6.1.1 BEM naming convention

With the advent of React and other sophisticated apps, developers quickly found that chaotically creating and using thousands of CSS classes and IDs made CSS specificity very hard to figure out, and easy to introduce design regressions. Thus, they came up with rules to keep naming more clear and consistent. The most popular is BEM ("Block Element Modifier").

In BEM, we ONLY use classes to style, and we think of 3 different classes: "Block" (highest level, we divide our page into these), "Element" (something more minor within the Blocks), and "Modifier" (for styling variants of Blocks or Elements).

```
.NavBar { }           /* Refers to the highest level ("Block" in BEM) page element */
.NavBar-navLink { }   /* Refers to something less important inside of NavBar ("Element" in BEM)*/
.NavBar--sticky { }   /* Refers to a varient of NavBar ("Modifier" in BEM)*/
```

You have a couple of options to read more on BEM:

1. BEM-based SUIT naming convention (most popular these days, shown above, and used in solutions and activities):

github.com/suitcss/suit/blob/master/doc/naming-conventions.md

2. The original BEM naming documentation (best explanation, has diagrams):

getbem.com/introduction/

Appendix: Alternative Homework

Unlike regular homework assignments, "Alternative Homework" assignments do not contribute to a cumulative solo project, but instead are one-off assignments.

They are typically as hard or harder than the solo project homework. They also typically have less "room for creativity" – you are given a single task with less flexibility in interpretation.

Why do alternative homework?

- Be better prepared for the interview process. Often you will have "take-home" interviews that involve making similar one-off projects from scratch.
- Extra reinforcement of the concepts of each week.
- Flesh out your GitHub profile with more repos. You want at least 10+ repos, so doing these extra homeworks will help with this sort of volume.
- **For students seeking career change, it is recommended you attempt to fully complete both homeworks.**

Grading

Alternate homeworks are optional, but should still be turned in to receive feedback.

NOTE: You may request to have the alternative homework graded and apply to your grade instead of the solo project homework, but it is recommended you attempt to fully complete the solo project homework first.