Human-Computer Interaction Design

COGS120/CSE170 - "Intro. HCI"

Instructor: Philip Guo

Lab 2 - Styling and publishing your website (2016-10-06)

by Michael Bernstein, Scott Klemmer, and Philip Guo

[Announce on Piazza] Before coming to lab 2:

1. Sign up for a Heroku account (if you don't have one), choosing "Node.js" as primary development language: https://www.heroku.com/

2. Install Node.js: https://nodejs.org/en/download/

Wi-fi may be slow in the classroom, so please download and install everything beforehand.

Since this is not a coding class, you will *not* turn in labs for credit. However, concepts from labs will be tested on Exam 1 and Exam 2.

Let's try a new format for labs:

1. I’ll give a mini-lecture on the slides for 20-30 minutes. You can follow along or jump ahead at your own pace. [room will be mostly silent so that everyone can hear clearly]

2. Then we will open it up for free-form lab work with TAs walking around to help. You can also come up to the podium to ask me for help too. [room will be louder]

Review: important commands you've used

- `ls` list what's in the current directory
- `cd <dir>` enter `<dir>`, or `cd ..` to go up to parent directory
- `git clone http://github.com/<YOUR USERNAME>/reponame.git` download a copy of the GitHub repo to your hard drive
- `git pull` get the most recent version of code from GitHub
- `git status` see which files have changed
- `git add <file>` pay attention to changes in `<file>` for committing
- `git commit -m "message"` commit (save) changes locally
- `git push` push all local commits to GitHub in the cloud

1) Let’s make your web site look good.

Lay out your content: HTML
Add styling: CSS
Leverage style frameworks: Bootstrap

2) Let’s publish your site online for everyone to see.

Develop your site locally (Lab 1): Mac, Windows, Linux
Publish to a server in cloud (Lab 2): Heroku (free web hosting service)

Setup: getting the files from GitHub

Fork the Lab 2 repository

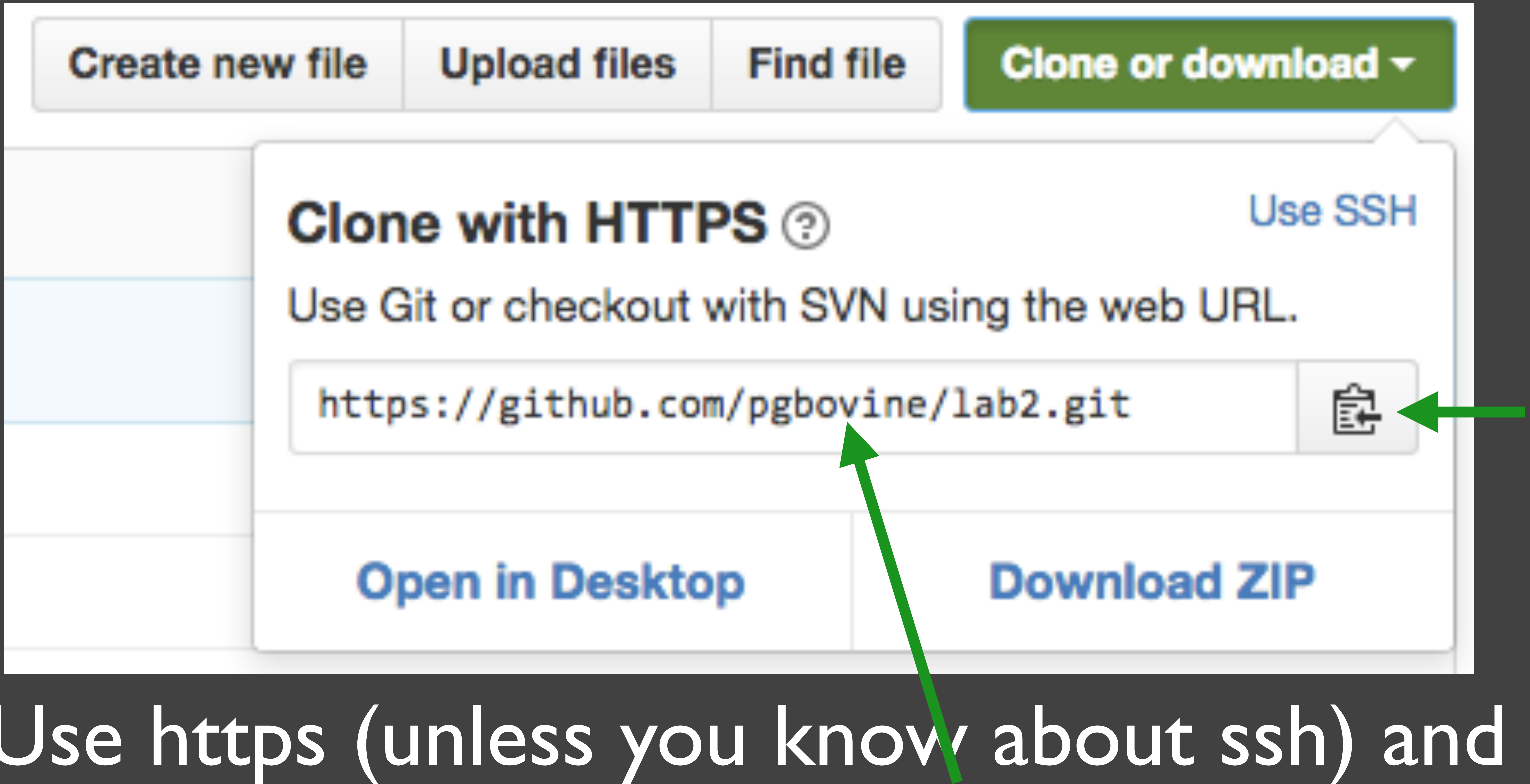
- Fork the repository at https://github.com/pgbovine/lab2 to make a copy into:
- https://github.com/<YOUR USERNAME>/lab2

pgbovine / lab2
forked from IntroHCI/lab2
Unwatch
1
Star
0
Fork
856

Open your terminal to get command line

- `cd` to the introHCI directory where you did Lab 1
- `ls` to make sure you see `lab1/` which you worked on last week

Clone your forked repo (not the original pgbovine one) into your introHCI directory



Use Git or checkout with SVN using the web URL.

```
https://github.com/pgbovine/lab2.git
```

Use https (unless you know about ssh) and substitute in your GitHub username

Clone your forked repo (not the original pgbovine one) into your introHCI directory

Terminal
Shell
Edit
View
Window
Help
132°
5.6KB/s
1.1KB/s
introHCI — msbernst@dn0a23115b: ~/Documents/introHCI — ..ents/introHCI — zsh

```
➜  introHCI
```

Making your site look good by styling your content

Heading 1
paragraph
div box (invisible)
image
hyperlink

```
<body>
    <div>
        <!-- h1 through h6 are headers. The higher the number, the s
        <!-- p means paragraph -->
        <div>
            <h1>Michael Bernstein</h1>
            <p>human-computer interaction &middot; social computing
        </div>

        <h4>Projects</h4>
        <!-- divs are invisibl                      that stack vert
        <!-- img tags are imag                will deliver random im
        <!-- a are anchors, also known as hyperlinks. Use the href a
        <div>
            <a href="project.html">
                <img src="images/lorempixel.people.1.jpeg" alt="Lore
                <p>Waiting in Line</p>
            </a>
        </div>
        <div>
            <a href="project.html">
                <img src="images/lorempixel.cit           alt="Lorem
                <p>Needfinding</p>
            </a>
        </div>
        <div>
            <a href="project.html">
                <img src="images/lorempixel.technics.1.jpeg" alt="Lo
                <p>Prototyping</p>
            </a>
        </div>
        <div>
            <a href="project.html">
                <img src="images/lorempixel.abstract.1.jpeg" alt="Lo
```

HCI Project Portfolio

msb-lab2.herokuapp.com

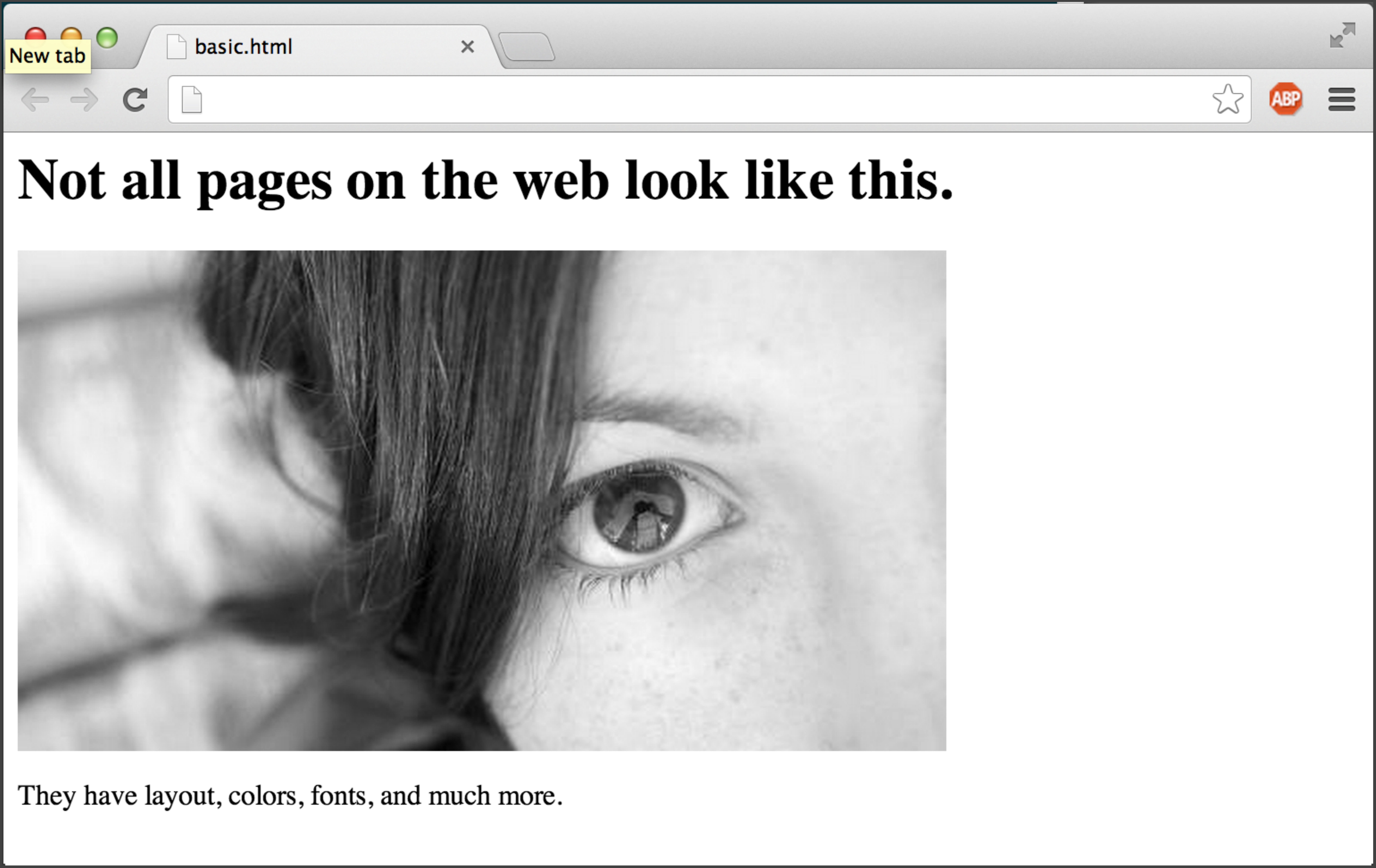
Michael Bernstein

human-computer interaction · social computing · crowdsourcing

Projects

Waiting in line

Add styling



Not all pages on the web look like this.

They have layout, colors, fonts, and much more.

The Big Idea

- Add **classes** to elements on the web page

```
<div class="project">
    <img class="thumbnail" src="img1.png" />
    <img class="thumbnail" src="img2.png" />
</div>
```

- Define the style for each class

```
.project {
    background-color: gray;
    margin-left: 10px;
}
```

Cascading Style Sheet

```
img .thumbnail {
    width: 50px;
    height: 100px;
    border: 1px solid #434343;
}

.contact-info {
    font-size: 10pt;
    color: #cccccc;
}

.project {
    background-color: gray;
    margin-left: 10px;
}
```

Open lab2/static/index.html in Sublime Text

- Add the link to the CSS file:
  `<link href="css/introHCI.css" rel="stylesheet" />`
- CSS imports go inside the `<head>` element

index.html

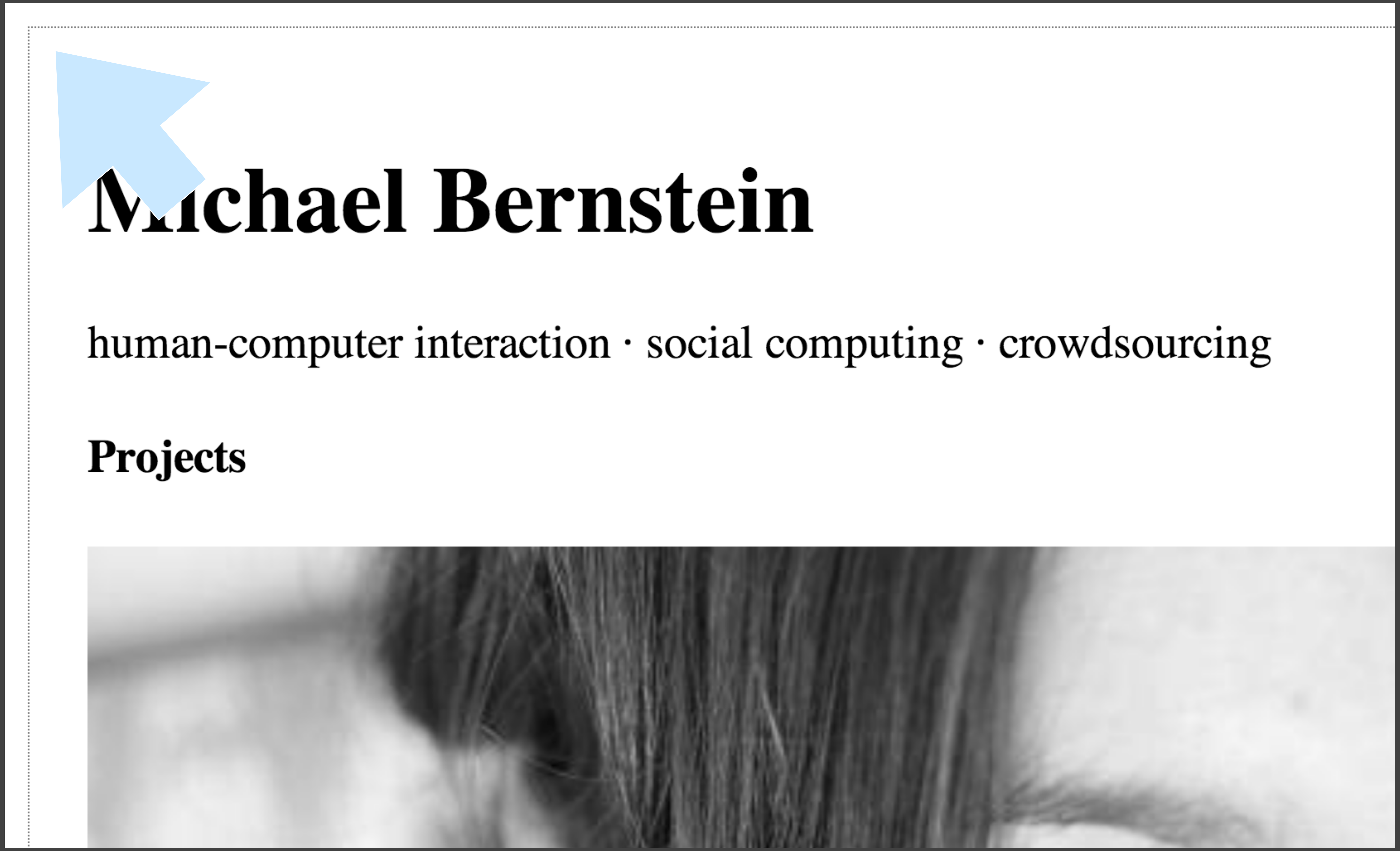
```
<!doctype html>

<html>
<head>
    <title>HCI Project Portfolio</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">


    <!-- this is a comment in HTML -->

    <!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media queries -->
    <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
    <!--[if lt IE 9]>
      <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>
      <script src="https://oss.maxcdn.com/libs/respond.js/1.3.0/respond.min.js"></script>
    <![endif]-->
```

Open static/index.html on your hard drive



Michael Bernstein

human-computer interaction · social computing · crowdsourcing

Projects

Keep this file open
in your browser!

Did it work?
Look for a gray dotted
box around the page

Add `class="project"` to the project divs

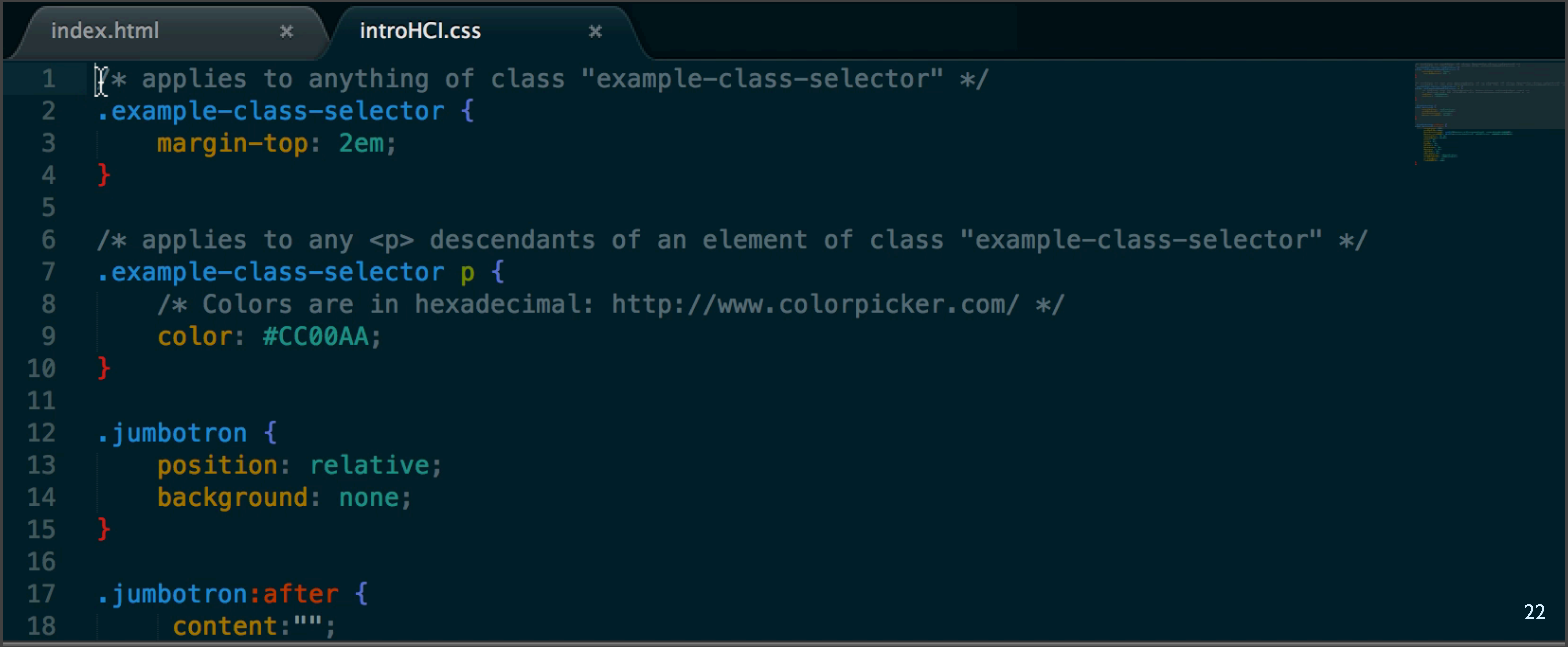
- For every `<div id="project1">`, `<div id=“project2">`, numbered 1 through 8, add `class="project"`
- Example: `<div class="project" id="project1">`

```
</div>


<h4>Projects</h4>
<!-- divs are invisible structuring elements that stack vertically by default. Use them to
organize your code -->
<!-- img tags are images. Lorem Pixel will deliver random images; handy! -->
<!-- a are anchors, also known as hyperlinks. Use the href attribute to tell the browser where
to go when the user clicks -->
<div id="project1">
    <a href="project.html">
        <img src="images/lorempixel.people.1.jpeg" alt="Lorem Pixel image">
        <p>Waiting in Line</p>
    </a>
</div>
<div id="project2">
    <a href="project.html">
        <img src="images/lorempixel.city.1.jpeg" alt="Lorem Pixel image">
```

Add the `project` selector to lab2/static/css/introHCI.css

- Give it `margin-left: 20px;`



index.html ✕ introHCI.css ✕

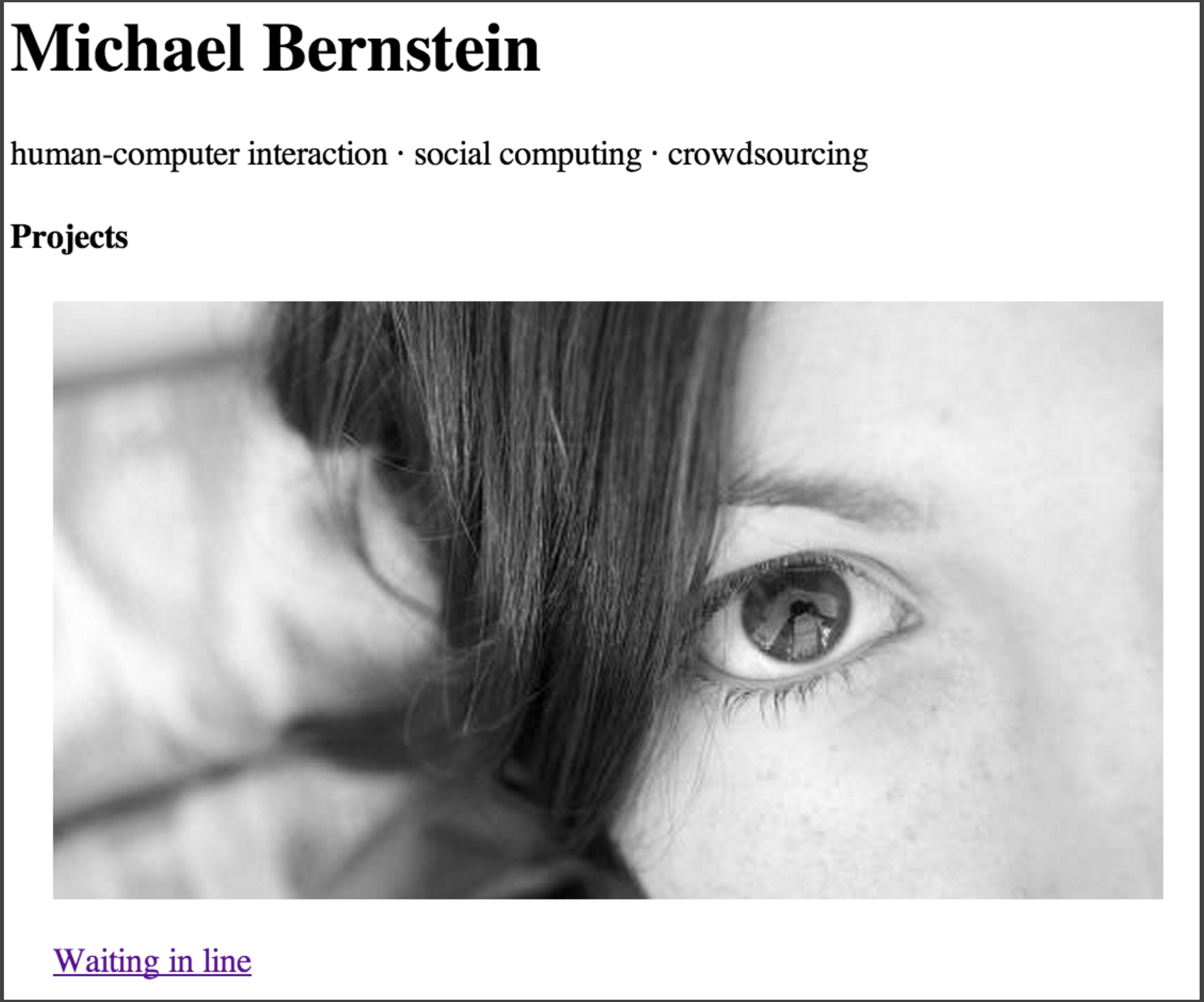
```css
/* applies to anything of class "example-class-selector" */
.example-class-selector {
    margin-top: 2em;
}

/* applies to any <p> descendants of an element of class "example-class-selector" */
.example-class-selector p {
    /* Colors are in hexadecimal: http://www.colorpicker.com/ */
    color: #CC00AA;
}

.jumbotron {
    position: relative;
    background: none;
}

.jumbotron:after {
     content:"";
```

Reload lab2/static/index.html



Michael Bernstein

human-computer interaction · social computing · crowdsourcing

Projects

Waiting in line

Now, projects have a left margin

Wait, what? `project` vs `.project`

- Define the class name (no dot) in the HTML

```
<div class="project">
```

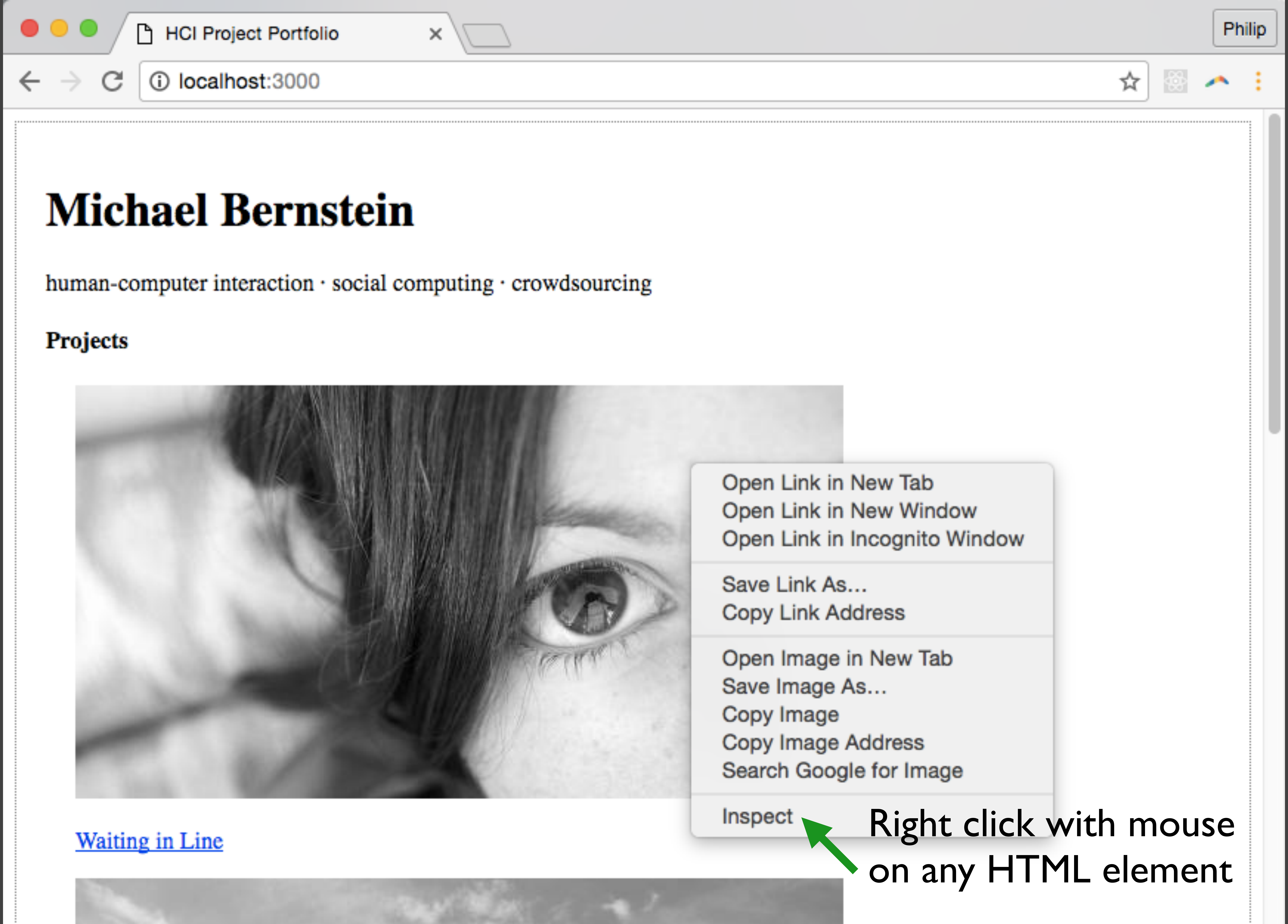
- The dot in CSS indicates we are searching for anything with that class name

```
.project { }
```

Your browser debugger lets you…

- See which properties are coming from which CSS file
- Edit the properties live
- Edit the HTML live

WARNING: all edits in debugger go away when you reload the webpage



HCI Project Portfolio

Philip

localhost:3000

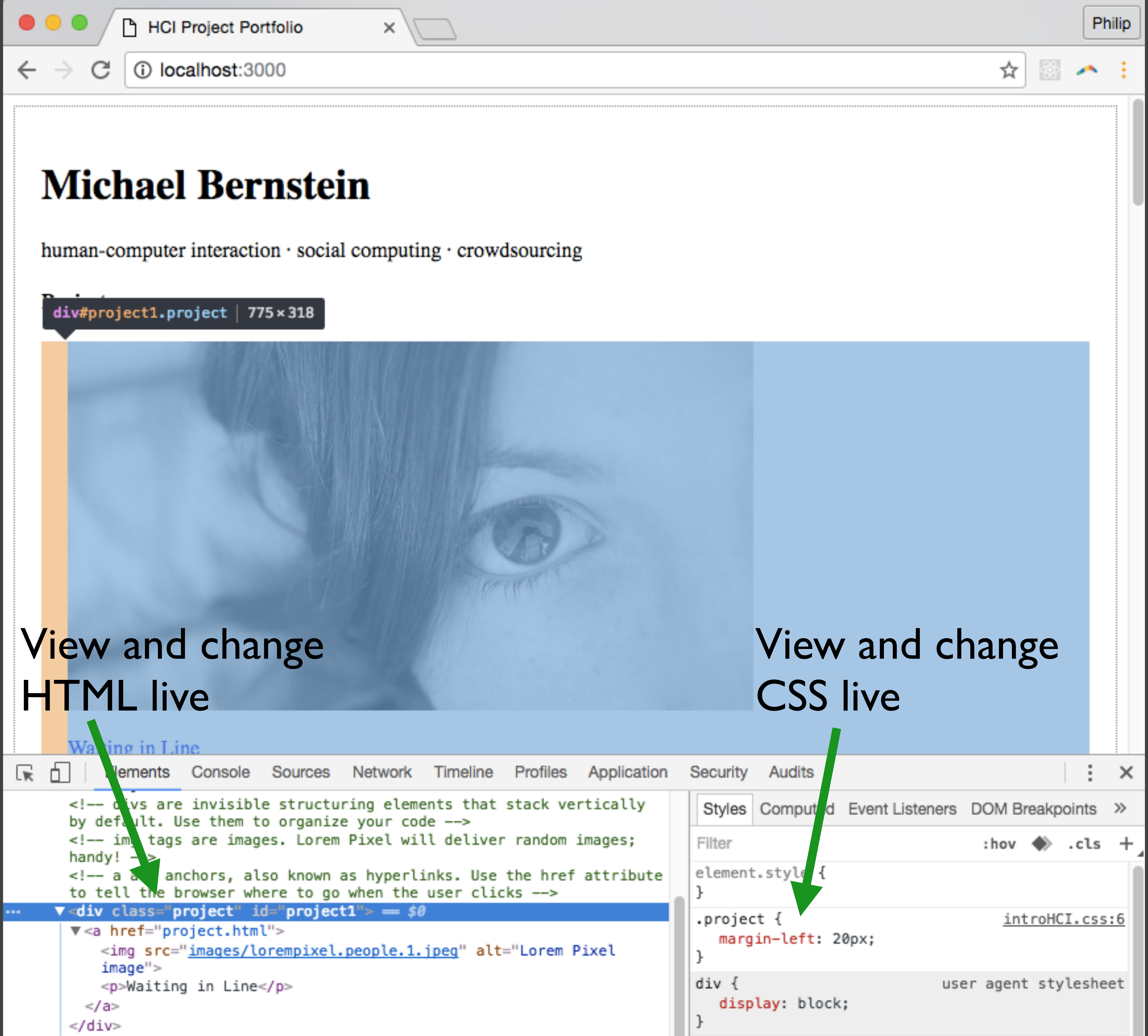
Michael Bernstein

human-computer interaction · social computing · crowdsourcing

Projects

Waiting in Line

Right click with mouse on any HTML element



Michael Bernstein

human-computer interaction · social computing · crowdsourcing

Waiting in Line

```
<!-- divs are invisible structuring elements that stack vertically by default. Use them to organize your code -->
<!-- img tags are images. Lorem Pixel will deliver random images; handy! -->
<!-- a are anchors, also known as hyperlinks. Use the href attribute to tell the browser where to go when the user clicks -->
▼<div class="project" id="project1"> == $0
  ▼<a href="project.html">
     <img src="images/lorempixel.people.1.jpeg" alt="Lorem Pixel image">
     <p>Waiting in Line</p>
   </a>
</div>
```

```
element.style {
}
.project {                          introHCI.css:6
  margin-left: 20px;
}
div {                      user agent stylesheet
  display: block;
}
```

View and change HTML live

View and change CSS live

Recommended debugger options

- Chrome: Developer Tools built-in
- Safari: Enable the Developer Tools
- Firefox: Developer Tools built-in
- Internet Explorer: Developer Tools built-in

But I want to style…

…all paragraph elements within a “project” div

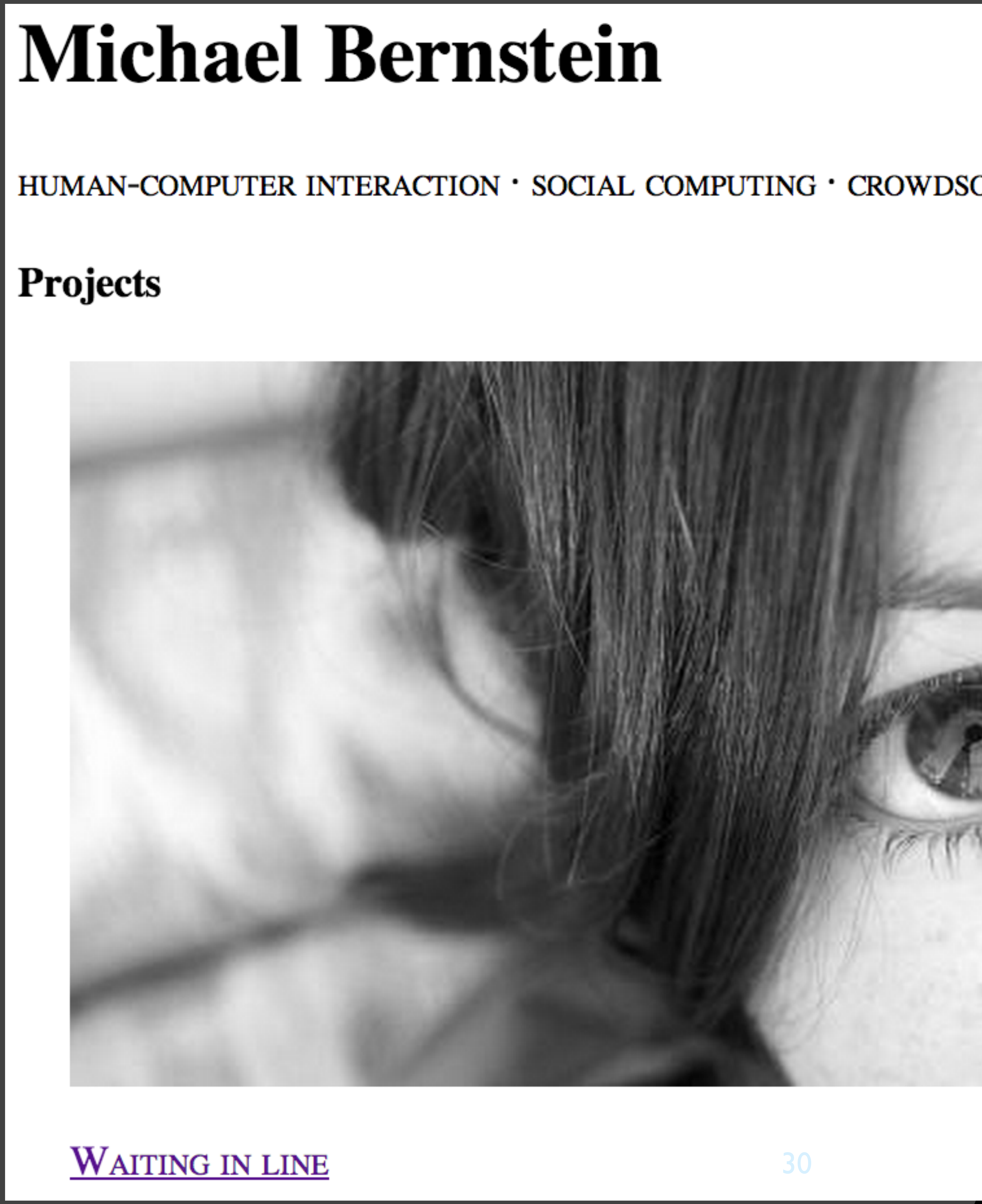
…all elements that have both classes “project” and “active”

Change your CSS selector

CSS selectors: basic element types

In introHCI.css —

```
p {
	font-variant: small-caps;
}
```



Michael Bernstein

HUMAN-COMPUTER INTERACTION · SOCIAL COMPUTING · CROWDSC

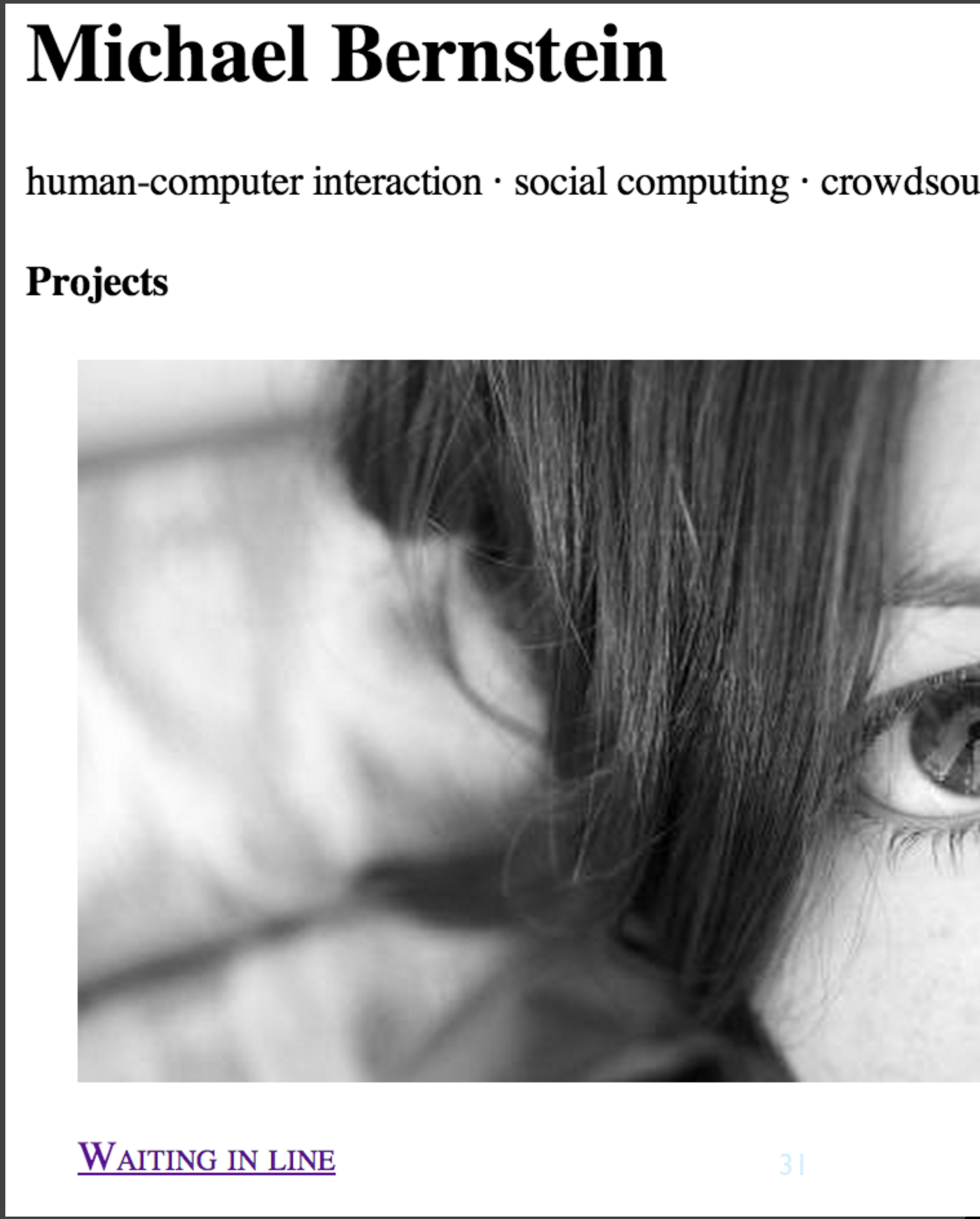
Projects

WAITING IN LINE

CSS selectors: descendants

In introHCI.css —

```
.project p {
     font-variant: small-caps;
}
```



Michael Bernstein

human-computer interaction · social computing · crowdsou

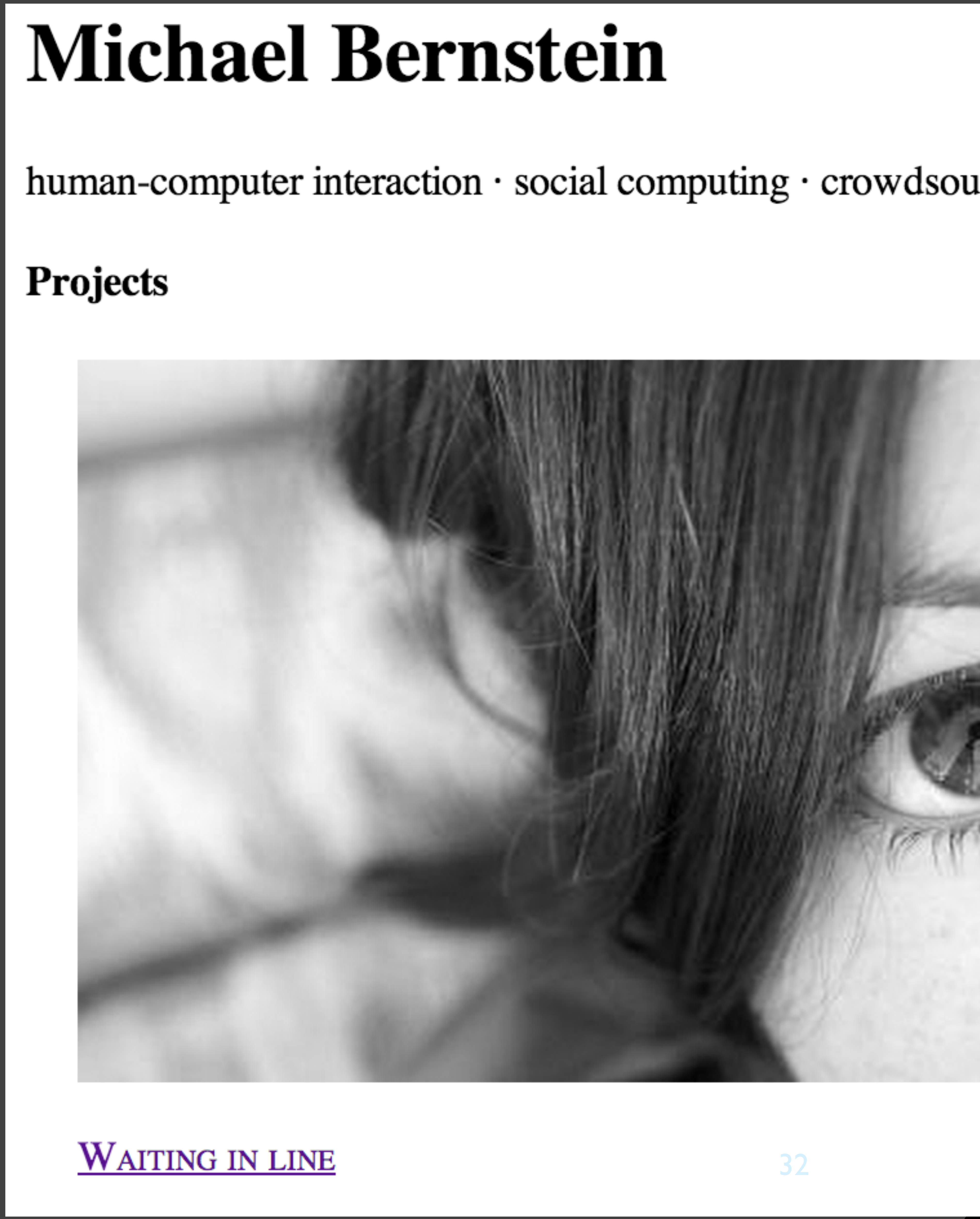
Projects

WAITING IN LINE

CSS selectors: ids (select specific elements)

In introHCI.css —

```
#project1 {
     font-variant: small-caps;
}
```



Michael Bernstein

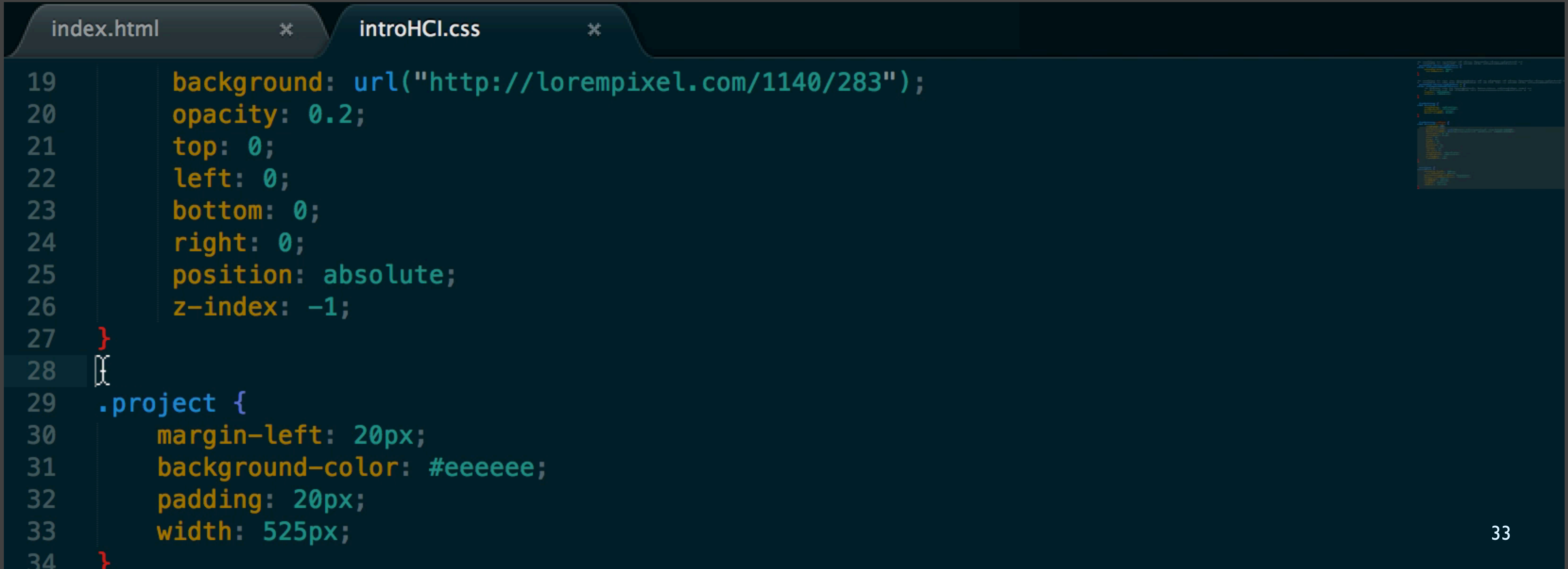
human-computer interaction · social computing · crowdsou

Projects

WAITING IN LINE

Delete the old `.project` CSS

- We never liked you anyway:
- Deleting the old .project CSS makes sure steps to follow appear



```
        background: url("http://lorempixel.com/1140/283");
        opacity: 0.2;
        top: 0;
        left: 0;
        bottom: 0;
        right: 0;
        position: absolute;
        z-index: -1;
}

.project {
    margin-left: 20px;
    background-color: #eeeeee;
    padding: 20px;
    width: 525px;
}
```

Leverage style frameworks

Don’t reinvent the wheel.

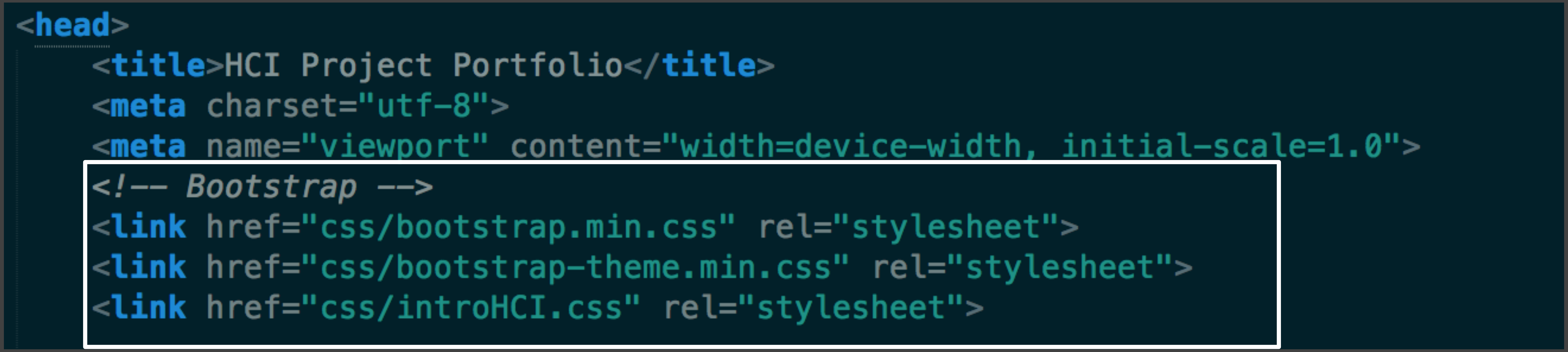
For most designs, you can copy and tweak.

Bootstrap

- Twitter's front-end web development framework
- Makes sane layout and styles easy to write
- Comes with predefined styles that you apply using their CSS style classes

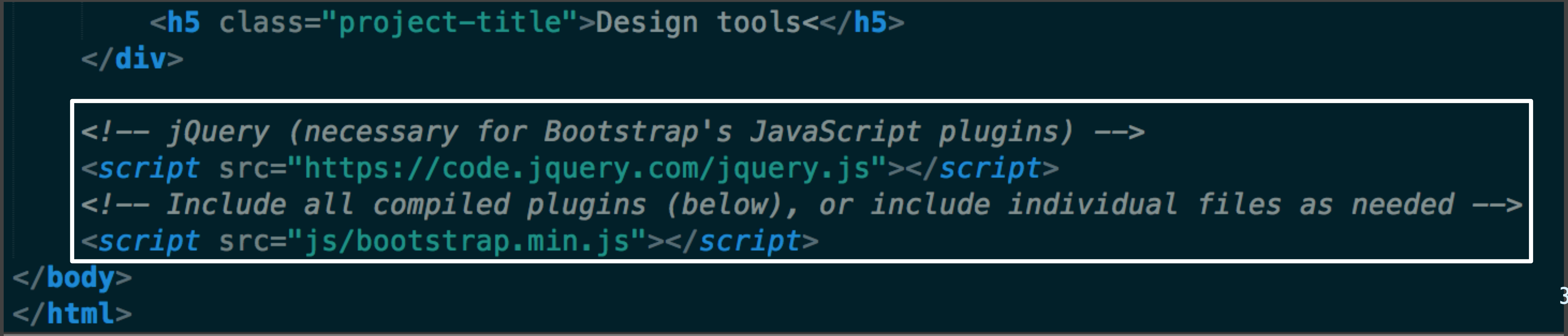
Add Bootstrap to your HTML

1. In <head>, **before** introHCI.css



```
<head>
    <title>HCI Project Portfolio</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!-- Bootstrap -->
    <link href="css/bootstrap.min.css" rel="stylesheet">
    <link href="css/bootstrap-theme.min.css" rel="stylesheet">
    <link href="css/introHCI.css" rel="stylesheet">
```

2. Just above </body> at the end of the document

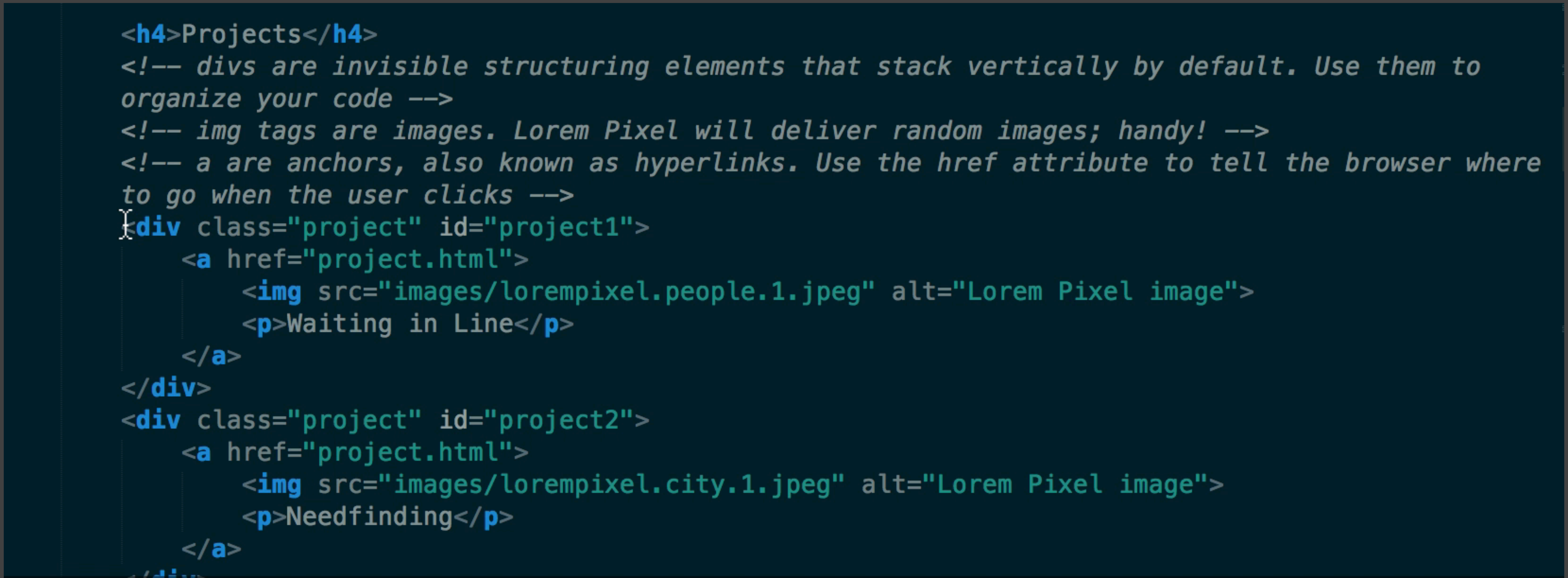


```
        <h5 class="project-title">Design tools<</h5>
    </div>

    <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
    <script src="https://code.jquery.com/jquery.js"></script>
    <!-- Include all compiled plugins (below), or include individual files as needed -->
    <script src="js/bootstrap.min.js"></script>
</body>
</html>
```

Add Bootstrap's styles to your HTML

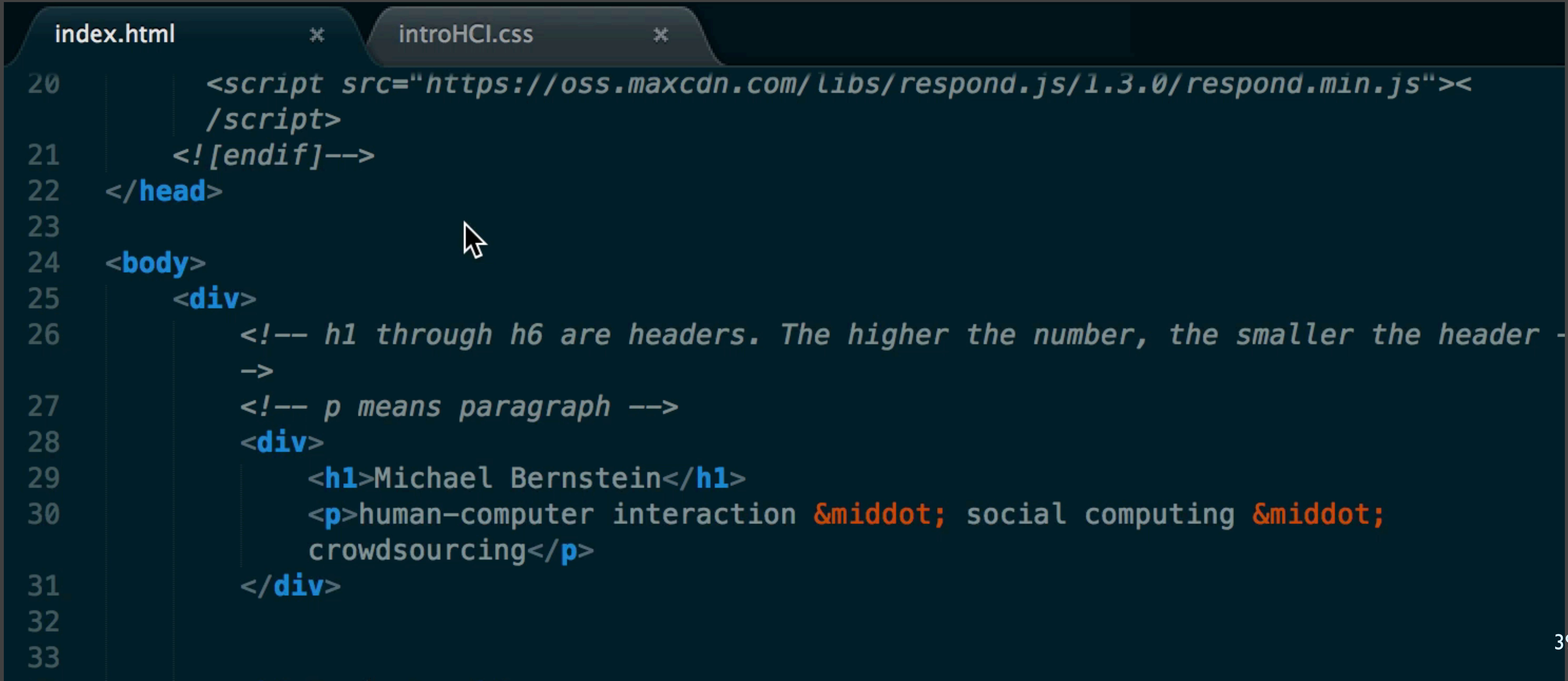
- Add the `thumbnail` class to the links:
  `<a href="project.html" class="thumbnail">`
- Add the `img` class to the images
  `<img src="images/lorempixel.foo.1.jpeg" alt="Lorem Pixel image" class="img">`



```
<h4>Projects</h4>
<!-- divs are invisible structuring elements that stack vertically by default. Use them to
organize your code -->
<!-- img tags are images. Lorem Pixel will deliver random images; handy! -->
<!-- a are anchors, also known as hyperlinks. Use the href attribute to tell the browser where
to go when the user clicks -->
<div class="project" id="project1">
    <a href="project.html">
        <img src="images/lorempixel.people.1.jpeg" alt="Lorem Pixel image">
        <p>Waiting in Line</p>
    </a>
</div>
<div class="project" id="project2">
    <a href="project.html">
        <img src="images/lorempixel.city.1.jpeg" alt="Lorem Pixel image">
        <p>Needfinding</p>
    </a>
```

Add Bootstrap's styles to your HTML

- Put the entire `body` inside a `<div class="container">`
- Containers add margins



```
        <script src="https://oss.maxcdn.com/libs/respond.js/1.3.0/respond.min.js"><
        /script>
    <![endif]-->
</head>

<body>
    <div>
        <!-- h1 through h6 are headers. The higher the number, the smaller the header -
        ->
        <!-- p means paragraph -->
        <div>
            <h1>Michael Bernstein</h1>
            <p>human-computer interaction &middot; social computing &middot;
            crowdsourcing</p>
        </div>


```

Michael Bernstein

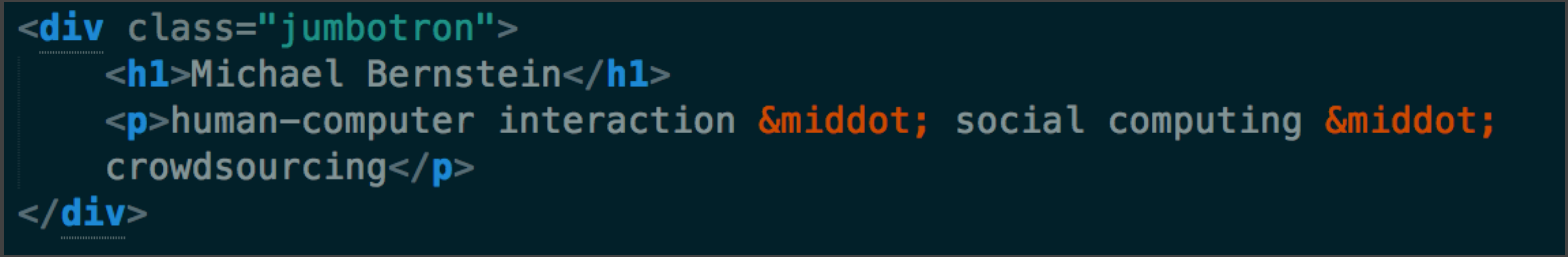
human-computer interaction · social computing · crowdsourcing

Projects

Waiting in Line

Lookin' good

- Add class jumbotron to the header



```
<div class="jumbotron">
    <h1>Michael Bernstein</h1>
    <p>human-computer interaction &middot; social computing &middot;
    crowdsourcing</p>
</div>
```

- The jumbotron is built into Bootstrap, and we have tweaked it in the CSS that came with the lab
- What does it do? Display the contents in big fonts and also add a randomly-generated background image from:
  - http://lorempixel.com/1140/283

Stretch goal

Create *custom* styling for the lab 2 portfolio page

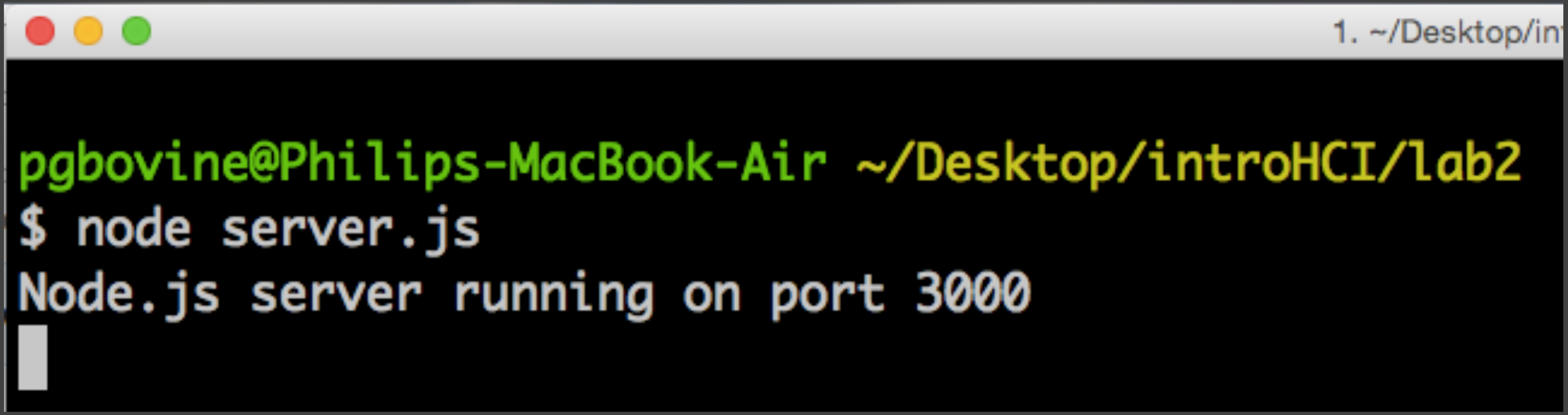
2) Let's publish your portfolio website.

Develop your site locally (Lab 1): Mac, Windows, Linux
Publish to a server in cloud (Lab 2): Heroku (free web hosting service)

NOTE: If you cannot get Node.js working, that’s fine for now … you can get help setting it up later. Jump ahead to “Publishing your web site”

Start Node.js web server at http://localhost:3000

- Command: `node <filename>`
- Change directory (`cd`) into `introHCI/lab2/`
- Run in that directory: `node server.js`
- Development servers (http://localhost) are only accessible on your own computer. This one runs on port 3000.

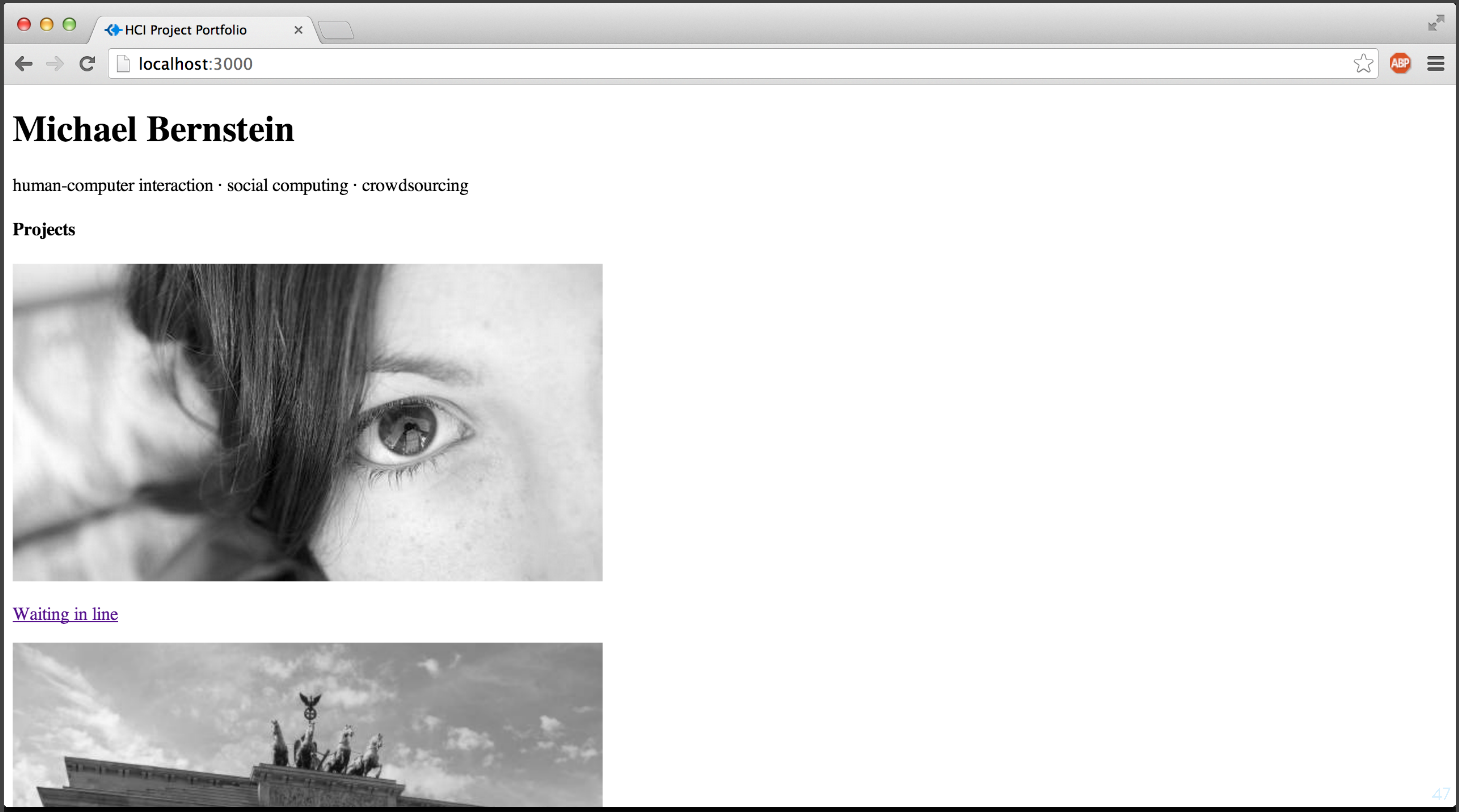


```
pgbovine@Philips-MacBook-Air ~/Desktop/introHCI/lab2
$ node server.js
Node.js server running on port 3000
```

Mac/Linux users can use any terminal app

Windows users can use the command prompt app or PowerShell built into Windows. Windows help: http://blog.teamtreehouse.com/install-node-js-npm-windows

Working! Note the URL is localhost:3000



Michael Bernstein

human-computer interaction · social computing · crowdsourcing

Projects

Waiting in line

Logs are back in your terminal window

```
Node.js server running on port 3000
10.0.2.2 - - [Tue, 07 Jan 2014 21:46:11 GMT] "GET / HTTP/1.1" 200 - "-" "Mozilla/5.0 (Macintosh;
 Intel Mac OS X 10_9_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/31.0.1650.63 Safari/537.36
"
10.0.2.2 - - [Tue, 07 Jan 2014 21:47:37 GMT] "GET / HTTP/1.1" 304 - "-" "Mozilla/5.0 (Macintosh;
 Intel Mac OS X 10_9_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/31.0.1650.63 Safari/537.36
"
10.0.2.2 - - [Tue, 07 Jan 2014 21:47:41 GMT] "GET /project.html HTTP/1.1" 404 - "http://localhos
t:3000/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_1) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/31.0.1650.63 Safari/537.36"
```

HTTP 200: OK — loaded the page
HTTP 304: Not modified — use the version you cached
HTTP 404: File not found — node.js doesn't know about that address

Quit the server

- Control-C to stop a running the server process.
  this will stop serving the page at http://localhost:3000

What is the difference between?

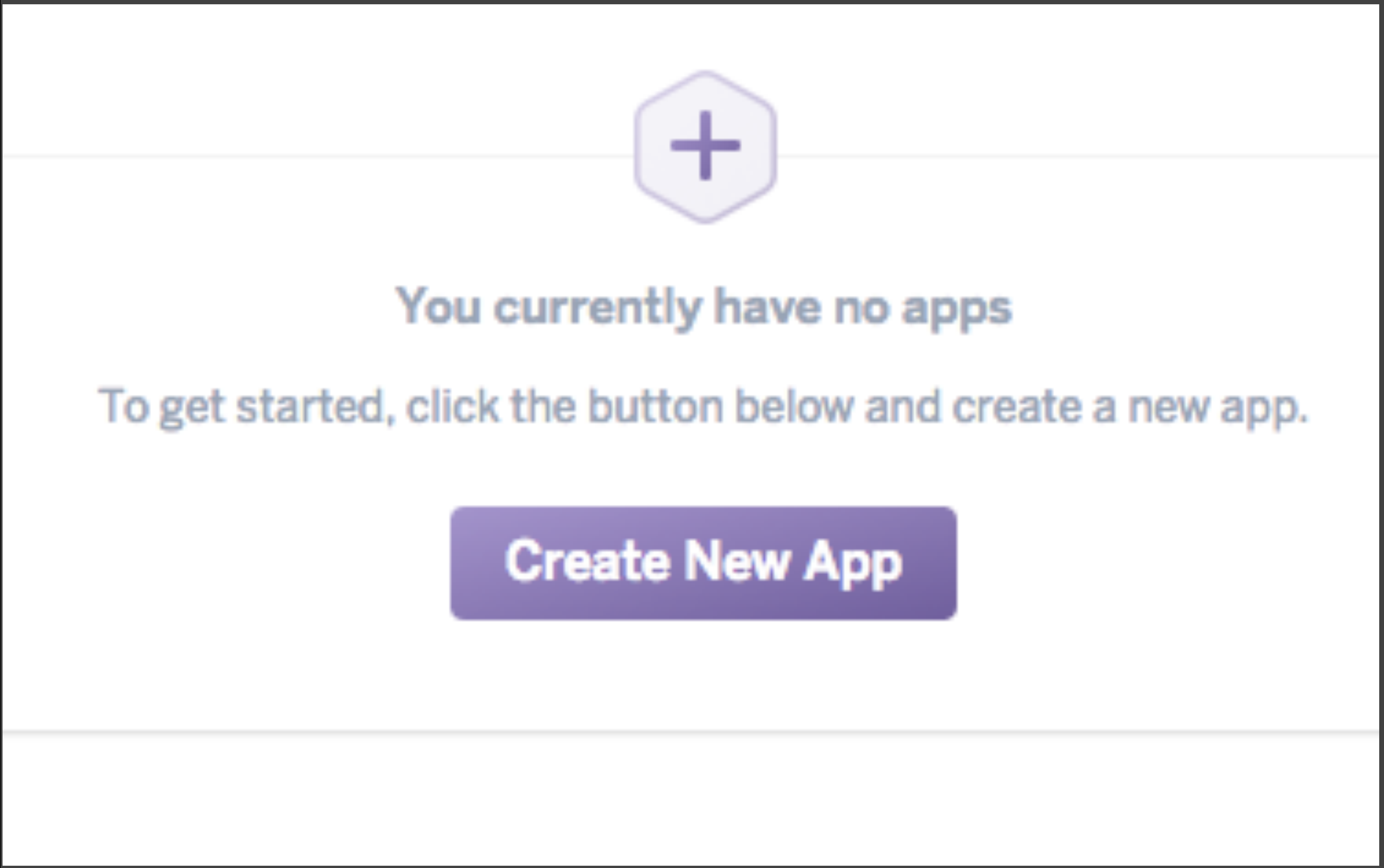
- a.) starting a Node.js server and loading your webpage through a local URL like: http://localhost:3000
- b.) simply loading your index.html webpage by opening the file on your computer?

- Answer: You need to start a server and load the webpage through a URL when you want to run server-side scripts to dynamically generate/load content for your web application of the sort you will create for your class project. However, if you simply have a static HTML/CSS/JavaScript webpage, then it doesn't matter which way you load your webpage.

Publishing your web site

We will use Heroku, which is a free online web hosting service that can run Node.js

Log into your Heroku account and create a new app


You currently have no apps

To get started, click the button below and create a new app.

Use your own username and not mine …

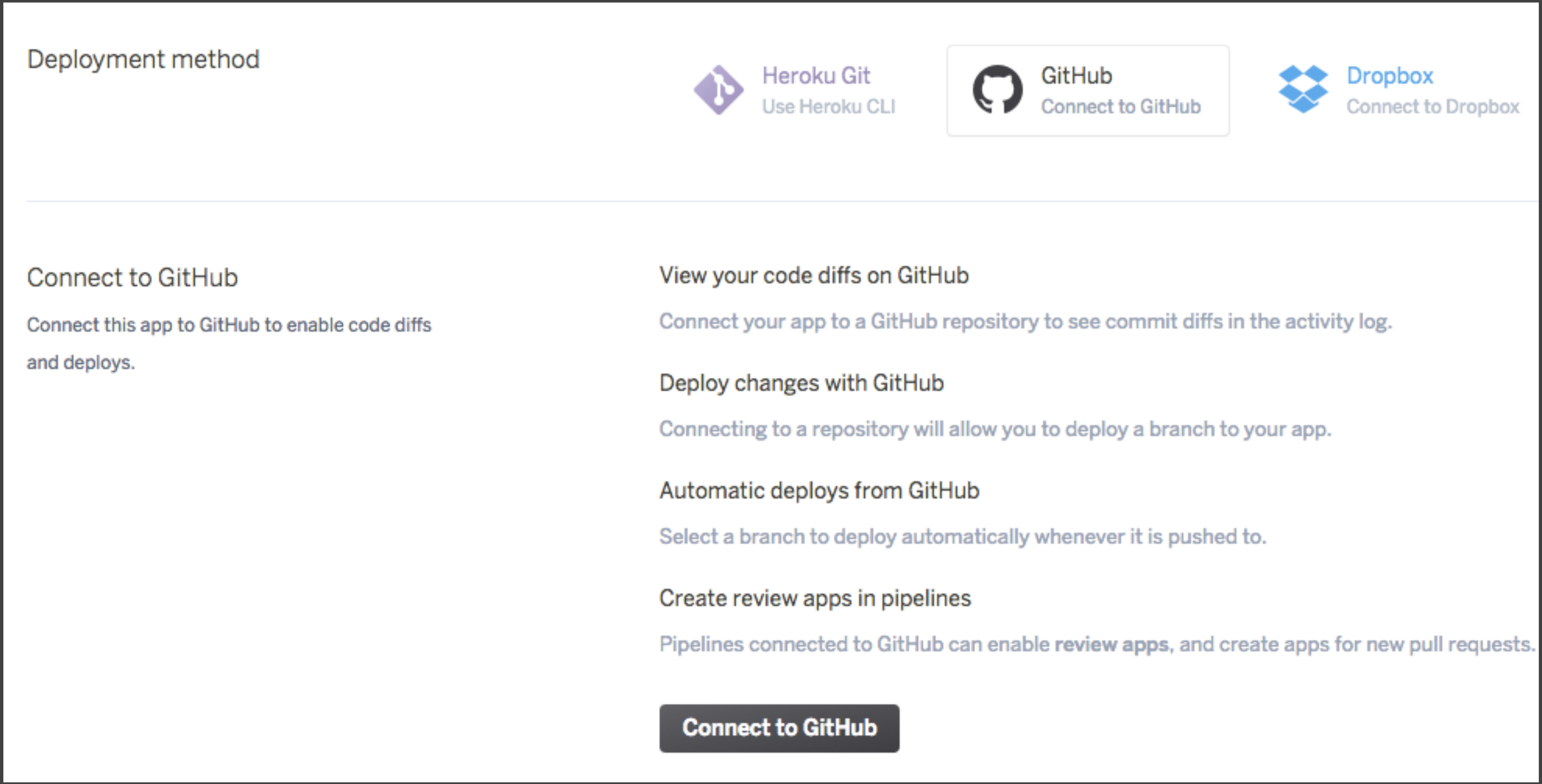
App Name (optional)

Leave blank and we'll choose one for you.

pgbovine-introhci-lab2

pgbovine-introhci-lab2 is available

Connect your Heroku app to your GitHub account



Deployment method

Connect to GitHub

Connect this app to GitHub to enable code diffs and deploys.

View your code diffs on GitHub

Connect your app to a GitHub repository to see commit diffs in the activity log.

Deploy changes with GitHub

Connecting to a repository will allow you to deploy a branch to your app.

Automatic deploys from GitHub

Select a branch to deploy automatically whenever it is pushed to.

Create review apps in pipelines

Pipelines connected to GitHub can enable **review apps**, and create apps for new pull requests.

Find the lab2 repository in your account to connect to

Connect to GitHub

Connect this app to GitHub to enable code diffs and deploys.

Search for a repository to connect to

pgbovine
lab2
Search

Missing an organization? Ensure Heroku Dashboard has organization access.

pgbovine/lab2

Connect

Deploy your GitHub master branch by clicking "Deploy Branch"

Manual deploy

Deploy the current state of a branch to this app.

Deploy a GitHub branch

This will deploy the current state of the branch you specify below. Learn more.

master
Deploy Branch

Receive code from GitHub ✓

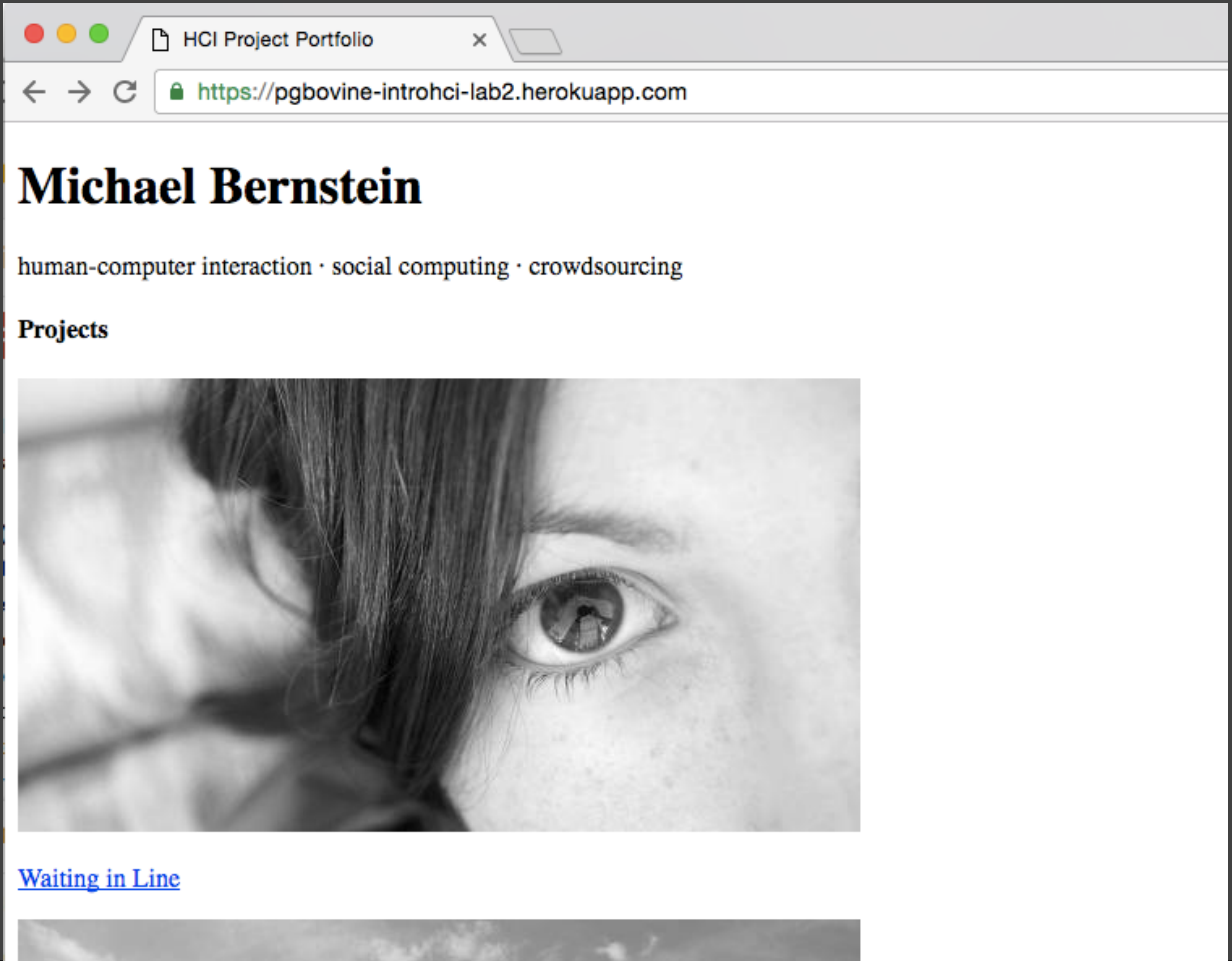
Build master Show build log ✓

Deploy to Heroku ✓

Your app was successfully deployed.

View

Your site is deployed (published) online!



HCI Project Portfolio

https://pgbovine-introhci-lab2.herokuapp.com

Michael Bernstein

human-computer interaction · social computing · crowdsourcing

Projects

Waiting in Line

When doing your assignments, we highly recommend creating *TWO* separate Heroku apps: one called “production” and the other called “development”. When submitting each assignment, deploy your site to the “production” app and ***leave that unchanged*** until the next assignment so that your TA can grade your submitted version. For your own personal testing and debugging, deploy to the “development” app.