WISE GUIDES
CAREER CHANGER

Rev. 25Jan2019

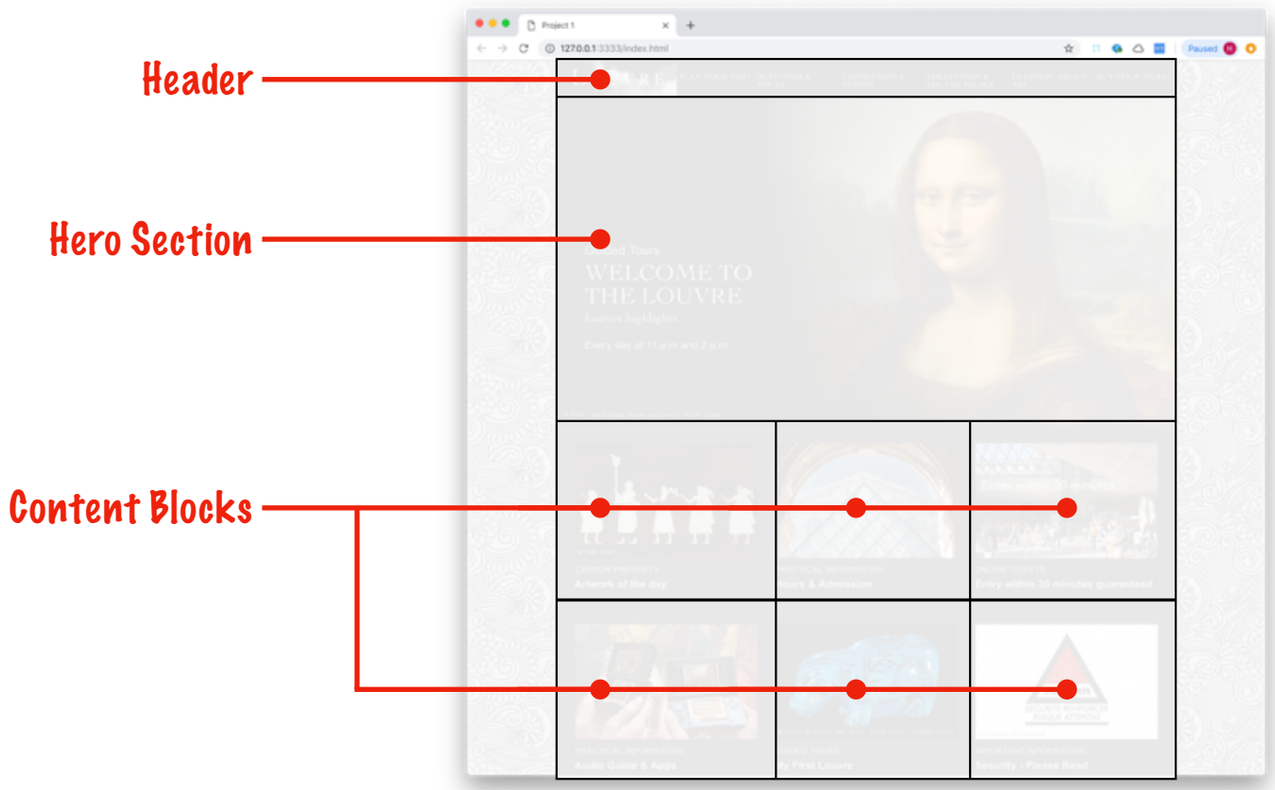
Exercise 0

UNDERSTANDING THE INDIVIDUAL ROLES OF HTML, CSS, AND JAVASCRIPT

Professional front-end programming largely centers around manipulating web pages using JavaScript. But in order to have a web page to manipulate, you first need to create a web page — and that means you'll need to learn about two complimentary technologies: HTML and CSS.

To understand how HTML, CSS, and JavaScript work together, think of a human body. The body has structure to it in the form of a skeleton. It's appearance is largely determined by the flesh around those bones, and then the body is alive, which gives it the ability to move and act.

If we use the body as an analog for a web page, HTML is the skeleton. It provides the structure to a web page. Here is a web page (from www.louvre.com) with the skeleton revealed as a set of named areas.



We use HTML to build that skeleton — that is, the individual portions that make up a web page.

HTML is an acronymn for *Hyper-Text Markup Language* and it consists of *elements* that can be read by both humans and the browser itself. Let's look at the *markup* for the page above:

```
<html>
  <header></header>
  <div class="hero"></div>
  <div class="content-blocks">
    <div class="content-block><div>
    <div class="content-block><div>
    <div class="content-block><div>
    <div class="content-block><div>
    <div class="content-block><div>
    <div class="content-block><div>
  </div>
</html>
```

What can we discover about HTML by looking at this *snippet*? Which of these statements would you feel moderately confident about asserting based purely on your initial observations?

- HTML elements are defined by text enclosed in angle brackets (tags).
- Tags are used in opening and closing pairs.
- Other tags can be nested between a previous element's opening and closing tags.
- All tags are lower-cased.
- Some tags have an attribute called `class` and some do not.
- Elements that are nested within a previous element's opening and closing tags are inset.

- HTML has only three elements: `<html>` `<header>` and `<div>`.
- Closing tags have the same format as opening tags — but are prefixed with a forward slash: `/`

PATTERN RECOGNITION

You may be thinking: "What are you asking *me* for? You're the teacher!" That's true — but you're taking this course not so that I can *teach*, but so that you can *learn*.

Students new to software development are typically panicked by their belief that this strange, new world requires some "programming gene" (that they're quite sure they don't have) and/or that it requires a MENSA-level intellect to master.

Let's explore that imaginary world further. The student imagines that the teacher has a vast theoretical understanding that they use to create things like *algorithms* and *functions* and *objects* and — well, as I said, the amount of knowledge required is vast.

Still in that world, you, the fledgling student, have the unenviable task of learning lots of arcane terminology and memorizing thousands of facts — and all the while, the sword of failure dangles above you, ready to kill your hope of becoming a professional programmer.

Say, did you ever play softball or volleyball — or, really, any sport? How'd you learn to do that? Pick up a book on *Mastering the Art of Softball*? I'm guessing not. Instead, you might have found yourself on a team — maybe in school, maybe with friends when you were young — and you sort of picked it up along the way.

Would that work with programming? Could you sort of "pick it up"? As it turns out, the answer is "Yes". Human brains aren't very good at storing huge amounts of data, but they're *really* good at spotting patterns. Within *days*, babies are able to recognize and differentiate different faces. By the time the baby is eight months old, they can recognize familiar faces from across the room. That kind of pattern recognition is still very difficult for computers to pull off.

In America, children begin their formal education around five years old. But by the time their anxious

parent sends the child off to their first day of school, the child can already speak. How did *that* happen? Learning a human language is remarkably difficult — much more so than learning a programming language — and yet children just sort of pick it up. Humans, it appears, are hard-wired for learning.

For your able-and-willing brain to recognize patterns, the key is *repetition*. We don't give pre-school children books on vocabulary and grammar; we do something much more effective: we speak to them over and over. And from that undifferentiated but large set of "language data", little brains recognize and internalize patterns.

The difficulty with the little quiz I gave you is that there's so little data upon which your brain can operate to spot patterns. Your brain will do its best with the scanty information you've been exposed to, but you're likely to have problems with (a) not seeing a pattern, and (b) seeing patterns that don't really exist.

Here are the answers to that quiz:

- ☑ HTML elements are defined by text enclosed in angle brackets (tags).
- ☑ Tags are used in opening and closing pairs.
- ☑ Other tags can be nested between a previous element's opening and closing tags.
- ☐ All tags are lower-cased.
- ☑ Some tags have an attribute called `class` and some do not.
- ☑ Elements that are nested within a previous element's opening and closing tags are inset.
- ☐ HTML has only three elements: `<html><header>` and `<div>`.
- ☑ Closing tags have the same format as opening tags — but are prefixed with a forward slash: `/`

Did you get one or two of them right? More? Less? Whether you got none of them right or all of them right, we can say two things: *you've begun learning* and *that learning is involuntary*. Like it or not, your brain is

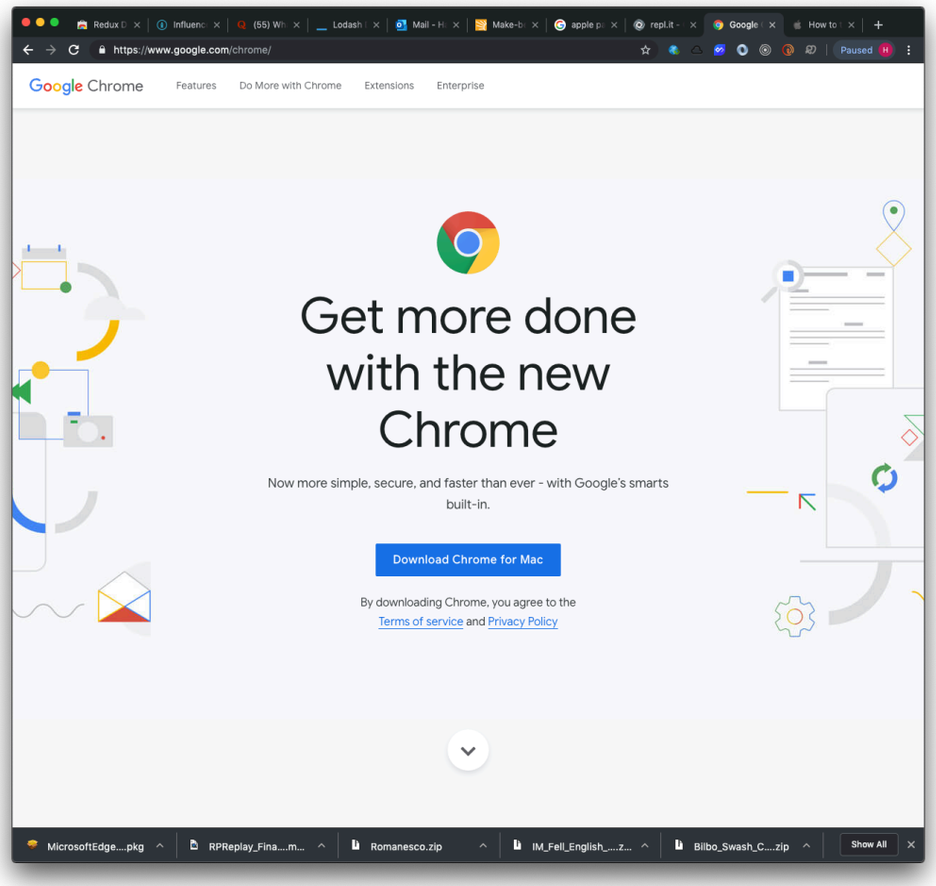
going to try to derive patterns even from the limited information I gave you. The job of the teacher is not to disseminate information so much as it is to repeatedly expose the student to situations that gently guide their brain to recognize correct patterns and to make useful connections between individual patterns.

SETTING UP A LEARNING ENVIRONMENT

In this course (and beginning with this project), you're going to communicate with computers using the same, natural skills you used when you learned your native language. But while learning your native language didn't require any special environment, learning a computer language will.

As the course progresses, you're going to learn how to set up and work in the same environment that professional programmers use, but for now, you'll use a simpler environment, preset to help you take baby steps.

For this, you'll need a modern browser. I recommend using Google's Chrome. If you don't already have it installed, point your current browser to: `https://www.google.com/chrome/`.



Since I'm using a Mac, the Chrome download screen shows instructions for downloading Chrome for Mac. If you're using Windows, your download screen will vary slightly.

Click the Download button and follow the instructions. Once you have it installed, point your Chrome browser to `https://repl.it/teacher/classrooms/136192/published`.

Click on the `index.html` tab. Right below this comment...

```
<!--vvv Place your code directly below this comment-->
```

... create 2 `<div>`s with a `class` of `outline`:

```
<div class="outline"></div>
<div class="outline"></div>
```

CSS

Back to the human body analogy, we said that HTML is analagous to the skeleton, providing structure to the web page. A human body with just a skeleton is the basis for many a horror movie. We don't want that, so let's look at the next technology you'll use to put "flesh" on those bones: CSS.

Cascading Stylesheets (abbreviated to CSS) lets us control the look of a web page. When you run the repl, you should see two boxes outlined in red. The boxes are the `<div>` elements you placed in the `index.html` page.

Why is the outline red—and not blue or orange or purple? Because I provided a file, `index.css`, that told the browser how I wanted all HTML elements with a `class` of `outline` to appear. You can go back to the repl site or, for your convenience, I've replicated it here:

```
.outline {
 min-height: 75px;
 outline: 2px solid red;
}
```

If you wanted the outline to be `blue` instead of `red`, what do you think you'd do to make that happen? Go back to `repl.it` and see if you can work that out.

Did you figure it out? You'd just change this code...

```
.outline {
 min-height: 75px;
 outline: 2px solid red;
}
```

to this:

```
.outline {
 min-height: 75px;
 outline: 2px solid blue;
}
```

I know it's tempting to just read along and not try out the code changes, but making the changes is part of that repetition that's so important to feed your brain so that it can spot patterns.

- [ ] And, if you wanted not a `solid` line, but a `dotted` one, what would you do? Try it in the repl.

Did you get it? You'd change this code:

```
.outline {
 min-height: 75px;
 outline: 2px solid blue;
}
```

to this:

```
.outline {
 min-height: 75px;
 outline: 2px dotted blue;
}
```

These are simple changes we're making with CSS, but as you lean more, you'll see just how impressive CSS can make your web pages look.

JAVASCRIPT

Back again to our human body analogy: so HTML is the skeleton and CSS is the flesh, but without something animating the body, we have a corpse. The third technology, JavaScript, provides life to the page by letting it interact with the user.

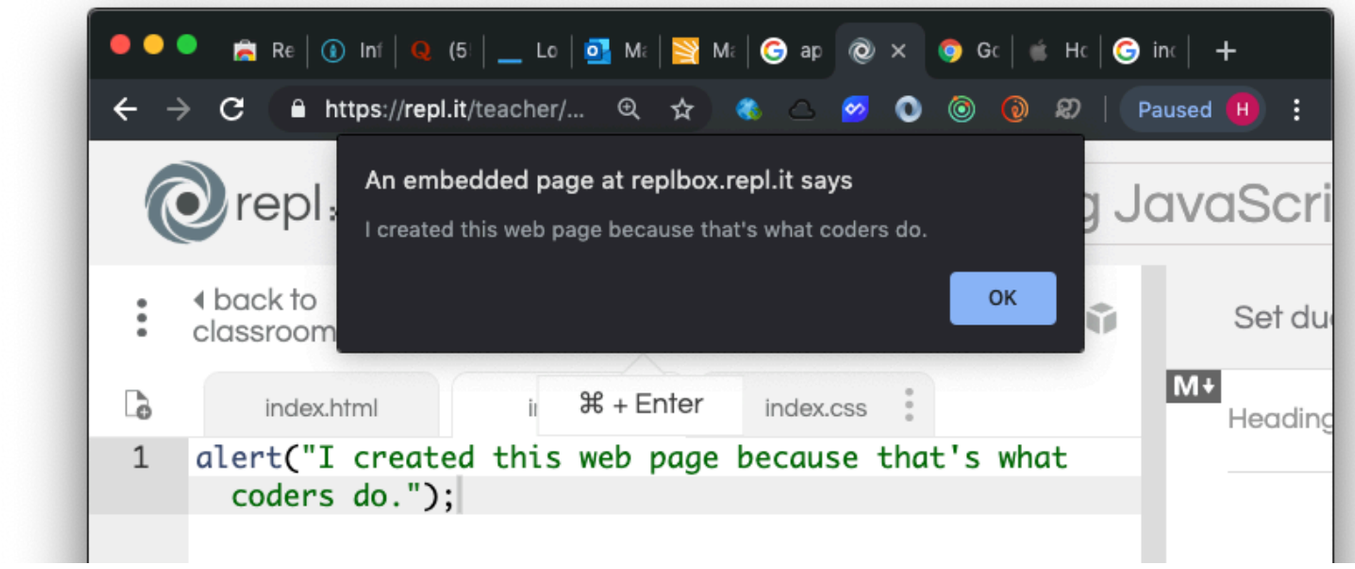
Let's get started with JavaScript. We're going to write some code that, when run, will pop up a window that says, "I created this web page because that's what coders do."

To do that, you'll create something called a JavaScript *function*. Functions are (usually) small bits of JavaScript that have some narrowly-defined purpose.

- Back in the repl, add this code to the `index.js` file:

```
alert("I created this web page because that what coders do.");
```

Now, when you click the **run** button, you should see this:



If you get stuck in any of these exercises, you can refer to the answer files. That will provide you with my solution that you can compare with yours.

- Now, let's try to make a change. Instead of saying `"I created..."`, use your name in place of the `"I"`. Does it work?

Notice that when you click the *alert box*'s **OK** button, the pop-up box goes away.

ABOUT THAT HTML...

You got the `<div>`s working. The CSS drew an outline around each `<div>`. Before we finish Exercise 0, let's do two more things. Back to `index.html`:

- Inside (between the opening and closing tags) of the first `<div>`, type the text, `Hello`.
- Inside the second `<div>`, type the text, `World`.

If you're wondering why this project is Exercise 0 instead of Exercise 1, it's an example of bad programmer humor.

When we number things in computer languages, we often start not with 1, but with 0. Thus this is Exercise 0.

(I warned you it was bad programmer humor...)

Now run the code again.

RECAP

In this, your first preschool exploration, you learned just a little about the three key technologies that you'll master in this course: HTML, CSS, and JavaScript. I used an example of the human body in which...

1. HTML is analagous to _____.
2. CSS is analgous to _____.
3. JavaScript is analagous to _____.

It's often the case that the first program you write in a new language simply outputs the text, "**Hello World**" to the screen. You just did that.

Welcome to the world of programming.