Task 1: Create an Initial Website Mockup

Background

The magic of web pages is that they use nothing more than plain text files. Plain text files have no fancy fonts, no colors, no pictures, no clickable links, and no animations. How then do web pages appear with all these things? With markup. Markup means that some of the text in a web page file is code that describes what to do with other text in the file. This code can say "make the following text be blue and bold" or "display the image that is in the following file."

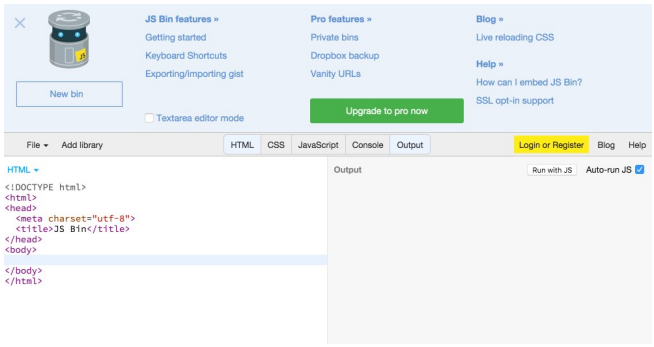
There are two important coding languages used to do this. **HTML** and **CSS**.

HTML stands for **Hypertext Markup Language**. **Hypertext** refers to the fact that HTML web pages can have links, i.e., text that when you click on it, you jump to another web page. This markup lets us **hyperlink** HTML files together into the **web** of documents we know so well today. HTML is used to label what different pieces of a content are for, such as titles, the sections, links, lists, tables, and images.

A Quick Demonstration

You can see for yourself how HTML works. There is a neat web site called **JS Bin** where you can create, run, and share little snippets of web page code.

Click on this link to open up the JS Bin main page.



Click the big X in the upper left to close the big banner. (If you want to get it back later, click on the small bin icon in the upper left.)

On the left, there is some HTML. This is boilerplate, i.e., HTML that every page has to have. The things between angle brackets, like `<head>` and `<body>`, are HTML instructions. They aren't displayed on a web page, but they can affect how things will look. You will learn what this HTML does very soon.

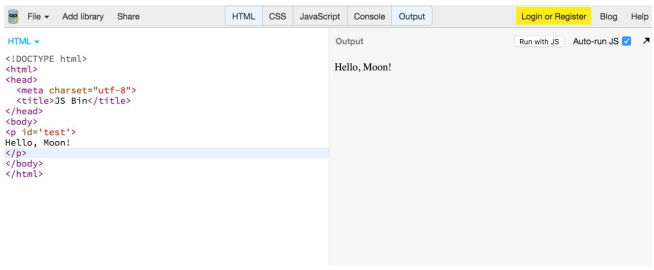
Let's add some HTML to show something.

```
<p id='test'>
Hello, Moon!
</p>
```

This is HTML code that says "make a paragraph (p) containing the text `'Hello, World!'` and label this paragraph `'test'`. You'll see what the label is for in a second.

Copy the lines of HTML code above, switch to JS Bin, click on the line between `<body>` and `</body>`, and paste the copied HTML.

As soon as you enter or change the HTML on the left, JS Bin displays the results that a browser would show on the right. Your screen should look like this now.

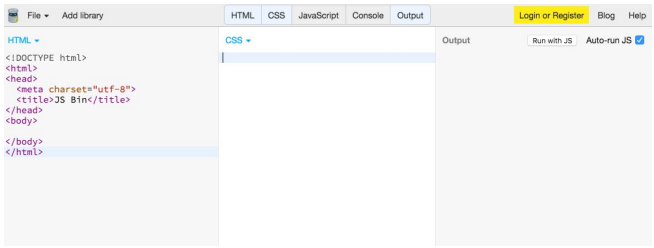


Congratulations! You just ran your first bit of working HTML code.

It is pretty plain though. HTML is all you need to make a readable web page, but it won't make a very pretty one. To make a page pretty, you use **CSS**.

CSS stands for **Cascading Style Sheets**. CSS is a language that lets you specify fonts, colors, indentation, positioning, table formats, and many other visual aspects of an HTML page.

Let's use CSS to make our HTML on JS Bin much more interesting. In JS Bin, click on the tab at the top that says **CSS**. This will open a box where you can enter CSS code. (You can open or close any JS Bin box by clicking on the appropriate tab.)



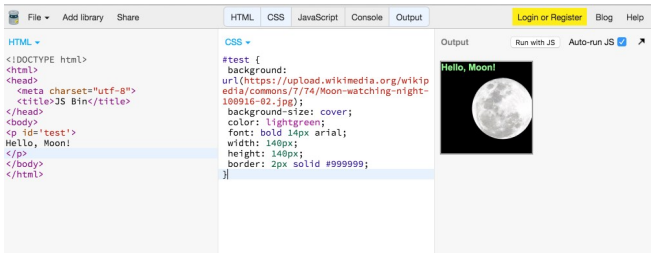
Copy the lines of CSS code below and paste them into the CSS box.

```
#test {
  background: url(https://c1.staticflickr.com/9/8500/8315697815_69cb2a99ba_b.jpg);
  background-size: cover;
  color: lightgreen;
  font: bold 14px arial;
  width: 140px;
  height: 140px;
  border: 2px solid #999999;
}
```

You'll learn soon enough what all these things mean. For now, here's a quick summary of what each line of this CSS is telling a web browser to do:

- `#test`: apply this styling information to the HTML item that has the ID `test`
- `background:` use a picture of the moon as the background; the picture is in a file at Flickr
- `background-size:` resize the picture to exactly cover the paragraph box
- `color:` make the text `light green` (here's a list of CSS color names)
- `font:` use a bold 14 pixel Arial font (a pixel is a standard tiny unit of size)
- `width` and `height`: make the paragraph box 140 pixels wide and high
- `border`: put a 2 pixel gray border around the paragraph

If you copied all of the above CSS correctly, JS Bin should now look like this:



Sweet! You've mixed text and graphics, using HTML and CSS.

Browsers, Servers and HTTP

The web works because of programs running on computers on the Internet send HTML and CSS files to each other, and display those files to you.

A **web browser** is a program you run on your computer that reads HTML and CSS and creates the visual display so familiar to web surfers. Web browsers are very complicated programs, but HTML and CSS files are comparatively quite simple.

A **web server** is a program running on some other computer that sends HTML and other types of files over a network to web browsers. It does this when it gets a request for a file. Those requests are themselves written in a language called **HTTP**, which stands for **HyperText Transfer Protocol**. You don't need to know what HTTP looks like to create web pages.

A web server and network connection are not required to see what an HTML file looks like. A web browser can open an HTML file you have on your own computer. This makes initial development and testing of web pages very easy.

A very useful tool when creating and testing web pages is an **HTML validator**. This is a program that checks your HTML for bad HTML syntax, from punctuation problems to the use of outdated HTML tags. The fact that a page looks OK in your browser is not a good test of your HTML. Browsers try to handle bad HTML, but they do so in different ways. A page with bad HTML might look fine in Firefox and completely broken in Internet Explorer. Fortunately, **the World Wide Web Consortium** (**W3C**) provides a free online HTML validator. See *Resources* for the link.

Required Resources

HTML

- ***Build Your Own Website The Right Way Using HTML & CSS*** by Ian Lloyd
  - Chapter 1 talks about the basic tools you'll need.

    - Chapter 2 shows an example HTML file. You can use it as a template for your web pages.
- **Learn to Code HTML & CSS: Develop & Style Websites** by Shay Howe
    - **TIP:**
      You'll come back to using CSS on the NYSL site in a later task.
    - Lessons discuss concepts first, then demonstrate them in practice through building a sample site.
    - Lesson 1 introduces HTML and CSS basics, text editors, and sets up the structure of a simple web page, including using CSS to style it (minimally).
    - Lesson 2 gets more into HTML, which you'll need to structure the NYSL site, and how to set up multiple pages, linked to each other.
    - Lesson 9 introduces adding images.

HTML Validator

- **The W3C HTML Validator** - This website will scan your HTML and report on everything incorrect, incomplete, or non-standard.
    - Because your web pages are not available on a server, you'll need to use the provided interactive form, where you can copy and paste your entire HTML code.

Web Browser

- You need at least one up-to-date version of one of the more popular browsers, such as Internet Explorer 9+, Mozilla Firefox 38+, Google Chrome 43+, Opera 20+, Safari 7+.

Text Editor

You can use any plain text editor you like, as long as you can tell it to insert spaces when you hit the TAB key. If you have no favorite, here are some recommendations.

Note: Do not use a word processor, like Word, WordPad, or OpenOffice Writer, and do not use a WYSIWYG HTML editor, like Dreamweaver. Similarly, if on Windows, do not use Notepad. Super-basic text editors like Notepad give you too little editing control over things like tab characters, and no support for writing syntactically valid HTML and CSS.

In whatever editor you use, **set the tab key to insert two spaces**. Two spaces is commonly used to indent nested HTML elements. It's enough to make the nesting clear, but not so much to push all the text over to the right.

MacOS

Unfortunately, TextEdit does not allow you to control the use of spaces when hitting the TAB key. We recommend you download and use the free TextWrangler editor, which is more powerful and custromizable.

- **TextWrangler** - Free download, many more features than TextEdit.
- **Tabs to Spaces in TextWrangler** - Shows you how to set the tab key to insert a certain number of spaces in TextWrangler. (Note: you have to restart TextWrangler after changing the preferences in order for the changes to take effect.)

Unix

- **Emacs** - Included or available with most Unix systems, somewhat complex to learn though.
- **Vim** - The other popular Unix editor.

Windows

- **Notepad++** - A popular replacement for Notepad with many features. (Note: Do not use Notepad)
    - **How to make tab insert spaces in Notepad++**

Additional Resources

HTML

- **Building Web Pages with HTML 5** - This resource provides some good guidance on HTML 5.
- Chapter 5 of *Build Your Own Website The Right Way Using HTML & CSS* - Goes into more detail on images.
- **W3Schools HTML Images Reference Page** - A resource that guides students in using images in HTML.

Image Editor

- **Specky Boy's List of Image Editors** - Lists image editors for all operating systems.

Zip File Help

To submit your working solution, you need to put your directory of files and folders, e.g., HTML, CSS, images, into one Zip file. Here are instructions for the three major platforms:

- Windows
- MacOS

- Linux