Creating Web Pages with HTML

Introduction

Web pages consist of text documents that contain plain text formatted with HTML (HyperText Markup Language) tags embedded within the text. HTML is a computer language used to format the content displayed in Web pages. The formatting consists of configuring the foreground and background color, adding white spaces between text, aligning text, configuring font, creating lists, tables, and forms. In this assignment we will learn how to use HTML to format plain text into Web pages.

Assignments in this course contain three main sections: ***Lab***, ***Tuiter***, and ***Challenge***. In the ***Lab*** sections we hand hold you through several exercises intended for you to practice various skills. The ***Lab*** section in this assignment will give you an opportunity to practice the concepts described in this assignment, i.e., HTML. Once you've had a chance to pratice with HTML, in the ***Tuiter*** section you'll be asked to apply what you've learned to building a Website similar of a popular social network site. The ***Challenge*** section explores additional, more challenging requirements that are optional for undergraduates, but required for graduate students.

Topics

- Creating Web content with the HyperText Markup Language (HTML)
- Formatting Web content with HTML tags
- Interacting with Web pages with HTML form tags
- Navigating between Web pages with HTML anchor tags

Lab

This section walks you through several exercises to familiarize yourself with HTML. Copy the examples into an HTML document as instructed and confirm that they render as intended. After practicing with the exercises you will be asked to apply the skills to create ***Tuiter*** on your own. Using ***IntelliJ***, open the React.js project you created in a previous assignment. From within IntelliJ, use ***File***, ***Open Project***, and navigate to the project directory, (***web-dev***), and click ***Open*** or ***OK***. From within IntelliJ, on the ***Project*** tab, open the ***web-dev*** directory, and then the ***public*** directory. Do all your work under the ***public*** directory of your project.

Heading Tags

Text documents are often broken up into several sections and subsections. Each section is usually prefaced with a short title or ***heading*** that attempts to summarize the topic of the section it precedes. The font of the section headings are usually larger and bolder than their subsection headings. This document uses headings to introduce topics such as HTML Documents, HTML Tags, Heading Tags, etc. HTML ***heading tags*** can be used to format plain text so that it renders in a browser as large headings. There are 6 heading tags: ***h1***, ***h2***, ***h3***, ***h4***, ***h5***, and ***h6***. Tag ***h1*** is the largest heading and ***h6*** is the smallest heading.

To practice using the heading tags we are going to create several headings and subheadings to introduce the topics we will cover in this assignment. Under the ***public*** directory, create directory ***labs/a2/html*** where you will practice several HTML exercises for assignment 2. Under the ***labs/a2/html*** directory, create an HTML file called ***index.html***. Copy the HTML below into the ***<body>*** tag of this new file.

Then, after the ***Heading Tags*** heading (highlighted in red here on the right), copy and paste the first paragraph of this section highlighted in yellow. To see the content of the Webpage, right click the ***index.html*** file and select ***Run index.html***.

HTML text

```
<h1>HTML Examples</h1>
<h2>Heading Tags</h2>
<!-- copy paragraph here -->
```

How the browser renders

HTML Examples

Heading Tags

The file will open in a browser window and the content should look similar to the content highlighted yellow at the beginning of this section. Note how the text surrounded by the ***<h1>*** tag is larger and bolder than the text surrounded by the ***<h2>*** tag, and both are larger than the text that has no tags arround it. The ***index.html*** document will be part of your deliverable and will contain the exercises that follow.

Paragraph Tag

Browsers ignore white spaces such as tabs and newlines. To add space between different paragraphs we can use the paragraph tag ***<p>***. Wrap text with the paragraph tag to add vertical spacing.

To practice using the paragraph tag, copy the code on the right at the end of the ***index.html***, but still within the ***<body>*** tag.

```
<h2>Paragraph Tag</h2>
<p>
      This is a paragraph. We often separate a long set of
      sentences with vertical spaces to make the text easier to
      read. Browsers ignore vertical white spaces and render all
      the text as one single set of sentences. To force the browser
      to add vertical spacing, wrap the paragraphs you want to
      separate with the paragraph tag
</p>
```

Below is another example of how the browser renders HTML text on the left column. The right column shows how the browser renders the HTML text on the left column. Note how the browser ignores line breaks and other white space formatting like tabs and content just flows from left to right and then wraps when there's no more horizontal space. This style of rendering is referred to as ***inline***. Inline content flows from left to right horizontally the whole width of its parent container and then wraps vertically when there's no more space.

HTML text

```
This is the first paragraph. The paragraph tag is used to
format vertical gaps between long pieces of text like this
one.

This is the second paragraph. Even though we added a
deliberate gap between the paragraph above and this
paragraph, by default browsers render them as one
contiguous piece of text as shown here on the right.
```

How the browser renders

This is the first paragraph. The paragraph tag is used to format vertical gaps between long pieces of text like this one. This is the second paragraph. Even though we added a deliberate gap between the paragraph above and this paragraph, by default browsers render them as one contiguous piece of text as shown here on the right. This is the third paragraph. Wrap each paragraph with the paragraph tag to tell browsers to render the gaps.

```
This is the third paragraph. Wrap each paragraph with the
paragraph tag to tell browsers to render the gaps.
```

Applying the paragraph tag below lets the browser know we want to keep the vertical spacing.

HTML

```
<p>
This is the first paragraph. The paragraph tag is used to format
vertical gaps between long pieces of text like this one.
</p>
<p>
This is the second paragraph. Even though there is a deliberate
white gap between the paragraph above and this paragraph, by
default browsers render them as one contiguous piece of text
as shown here on the right.
</p>
<p>
This is the third paragraph. Wrap each paragraph with the
paragraph tag to tell browsers to render the gaps.
</p>
```

How the browser renders

This is the first paragraph. The paragraph tag is used to format vertical gaps between long pieces of text like this one.

This is the second paragraph. Even though there is a deliberate white gap between the paragraph above and this paragraph, by default browsers render them as one contiguous piece of text as shown here on the right.

This is the third paragraph. Wrap each paragraph with the paragraph tag to tell browsers to render the gaps.

Copy the HTML above on the left to the end of the ***index.html*** document in the ***Paragrah Tag*** section. Remember to keep all your content within the ***body*** tag. Refresh the Webpage and confirm it renders as shown on the right. Note how the paragraphs are now spaced vertically from one another. Both the paragraph and heading tags add vertical space and we refer to this style of rendering as ***block***. By controlling the inline and block styles of laying out content, we can achieve all sorts of useful layouts.

List Tags

List tags are used to create lists of related items. There are two types of lists: ***ordered*** and ***unordered***.

Ordered List Tag

Ordered list tags are useful for listing items in a particular order. Here's a list of steps to join the 1% in 4 "easy" steps.

HTML text

```
<h2>List Tags</h2>
<h3>Ordered List Tag</h3>
How to join the 1%
1.   Get a masters in computer science
2.   Get a $100K+ starting salary job
3.   Marry a like-minded engineer
4.   Do great work for a decade
```

How the browser renders

List Tags

Ordered List Tag

How to join the 1% 1. Get a masters in computer science 2. Get a $100K+ starting salary job 3. Marry a like-minded engineer 4. Do great work for a decade

Note that in the HTML text on the left we explicitly wrote the numbers 1., 2., etc., but the nice formatting is lost when the browser renders it on the right. Instead of rendering a list of items, each in its own line, they are instead all rendered on the same line. To achieve the desired format we'll use the ordered list tag.

The ordered list tag actually consists of a pair of tags

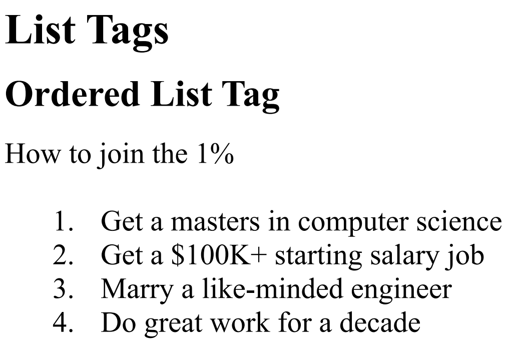
- <ol> declares the beginning of the list
- <li> declares an item in the list

Here's the same example from earlier, but now applying the ordered list tags to achieve the intended formatting.

HTML text

```
<h2>List Tags</h2>
<h3>Ordered List Tag</h3>
How to join the 1%
<ol>
        <li>Get a masters in computer science</li>
        <li>Get a $100K+ starting salary job</li>
        <li>Marry a like-minded engineer</li>
        <li>Do great work for a decade</li>
</ol>
```

How the browser renders



List Tags

Ordered List Tag

How to join the 1%

1. Get a masters in computer science
2. Get a $100K+ starting salary job
3. Marry a like-minded engineer
4. Do great work for a decade

Copy the HTML above to the end of ***index.html*** file and confirm it renders as shown on the right.

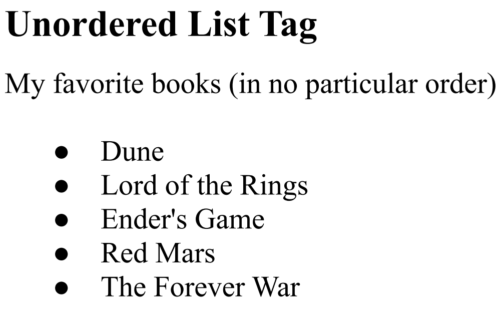
Unordered List Tag

The unordered list tag is similar to it's ordered version with the difference that the items are not numbered and instead bullets decorate each line item. The unordered list tag is ***<ul>***, but the list item tag is still ***<li>*** as shown below. Unordered lists are great for displaying a list of items in no particular order. Here's an example of an unordered list of my favorite books in no particular order.

HTML text

```
<h3>Unordered List Tag</h3>
My favorite books (in no particular order)
<ul>
        <li>Dune</li>
        <li>Lord of the Rings</li>
        <li>Ender's Game</li>
        <li>Red Mars</li>
        <li>The Forever War</li>
</ul>
```

How the browser renders



Unordered List Tag

My favorite books (in no particular order)

- Dune
- Lord of the Rings
- Ender's Game
- Red Mars
- The Forever War

Add the example HTML code above to the end of the ***index.html*** document to include it in your deliverable.

Table Tags

HTML began as a tool for sharing research results between scientists. These documents often consisted of data points captured as a result of some experiment. Each data point might have several attributes associated. An effective way to display or visualize these results were formatted in a data table with a row for each data point and a column for each attribute. The ***<table>*** tag allows formatting data into a table with rows and columns. For instance, consider capturing grade results for several quiz exams you might have taken over the semester. These might be captured using the following table.

Quiz	Topic	Date	Grade
Q1	HTML	2/3/21	85
Q2	CSS	2/10/21	90
Q3	JavaScript	2/17/21	95
Average			90

Several things to note:

1. The first row is formatted as headings for each column
2. There are 3 data points, one for each quiz, one in each row
3. Each data point has the same data types for each of the columns, e.g, Quiz, Topic, Date, Grade
4. The last row is formatted as a footer
5. The three first columns of the last row are merged into a single cell and unlike the 3 data rows

HTML ***table*** tag can be used to format the data with the following tags:

- ***table*** - declares the start of a table
- ***tr*** - declares the start of a row
- ***td*** - declares a table data cell
- ***thead*** - declares a row of headings
- ***tbody*** - declares the main data content rows of the table
- ***tfoot*** - declares a row as a footer
- ***th*** - declares a table cell as a heading

To practice using ***table*** tag, copy the HTML below to the end of index.html. The code implements the table shown earlier. You can ignore the comments on the right.

```
<h2>Table Tag</h2>                          <!-- declares the table, sets border and width -->
<table border="1" width="100%">             <!-- declares the table heading section -->
        <thead>                             <!-- declares the headings row -->
                <tr>                        <!-- declares heading for first column -->
                        <th>Quiz</th>       <!-- declares heading for second column -->
                        <th>Topic</th>      <!-- declares heading for third column -->
                        <th>Date</th>       <!-- declares heading for fourth column -->
                        <th>Grade</th>
                </tr>
        </thead>                            <!-- declares the table body section -->
        <tbody>                             <!-- declares the first row -->
                <tr>                        <!-- declares data for first row, first column -->
```

```
            <td>Q1</td>                     <!-- declares data for first row, second column -->
            <td>HTML</td>                   <!-- declares data for first row, third column -->
            <td>2/3/21</td>                 <!-- declares data for first row, fourth column -->
            <td>85</td>
        </tr>                               <!-- declares the second row -->
        <tr>                                <!-- declares data for second row, first column -->
            <td>Q2</td>                     <!-- declares data for second row, second column -->
            <td>CSS</td>                    <!-- declares data for second row, third column -->
            <td>2/10/21</td>                <!-- declares data for second row, fourth column -->
            <td>90</td>
        </tr>                               <!-- declares the third row -->
        <tr>                                <!-- declares data for third row, first column -->
            <td>Q3</td>                     <!-- declares data for third row, second column -->
            <td>JavaScript</td>             <!-- declares data for third row, third column -->
            <td>2/17/21</td>                <!-- declares data for third row, fourth column -->
            <td>95</td>
        </tr>
    </tbody>                                <!-- declares the table footer section -->
    <tfoot>                                 <!-- declares the footer row -->
        <tr>                                <!-- declares data spans 3 columns -->
            <td colspan="3">Average</td>    <!-- declares data for fourth column -->
            <td>90</td>
        </tr>
    </tfoot>
</table>
```

Image Tag

Use the image tag to render pictures in your HTML documents. The images can be anywhere on the internet, or a local image document in your Website.

```
<img src="my-picture.jpg"     <!--     Use img tag to embed pictures in HTML documents. The src
    width="200px"                      attributes references the image file either locally or remotely.
    height="300px"/>                   The width and height attributes configure the image size. If only
                                       width or height is provided, the other scales proportionally  -->
```

To practice using the image tag, copy the code below at the end of ***index.html***. The first image tag embeds an image from a remote server. The second one assumes you have a local image file called ***teslabot.jpg***. Search for Tesla Bot on the internet, and download an image that looks similar to the one shown below. Name the image ***teslabot*** keeping the original file extension.

```
<h2>Image tag</h2>

Loading an image from the internet:<br/>

<img
src="https://www.staradvertiser.com/wp-content/uploads/2
021/08/web1_Starship-gap2.jpg"
        width="400px"/>
<br/>

Loading a local image:<br/>

<img src="teslabot.jpg"
        height="200px"/>
```

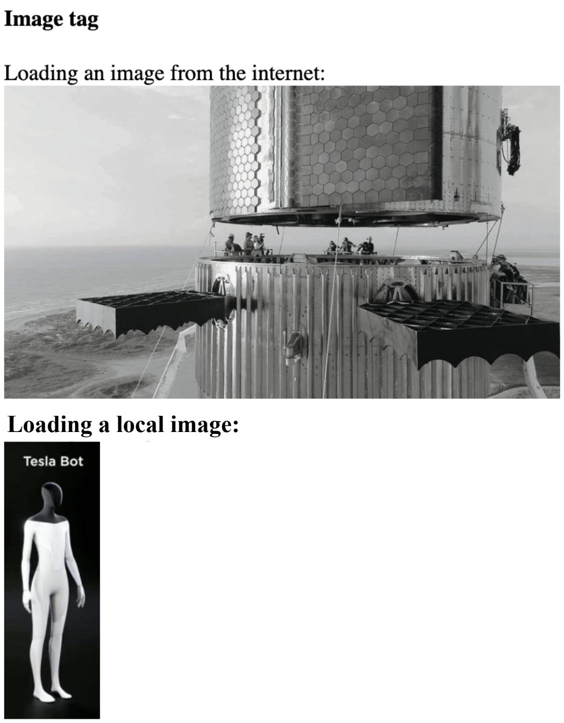


Image tag

Loading an image from the internet:

Loading a local image:

Form Tags

Form tags are useful for entering data. Let's take a look at the most common ones: ***form***, ***input***, ***select***, ***textarea***, ***radio***, ***checkbox***.

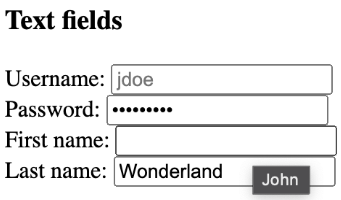
Text fields

Text fields are the most common of form elements allowing entering a single line of text.

```
<input  type="text"              <!--     use input tag's text type to declare a single line input field
        placeholder="hint"                text is default if type is left out. Use placeholder and title
        title="tooltip"                   to give a hint of what information you're expecting.
        value="COMEDY"/>                  Optionally initialize the value of the field with value attribute-->
```

To practice using text fields, add the following example at the end of ***index.html***. It creates a set of input fields for entering some personal information. The ***label*** tags below associate descriptive text with each form element. The is established by setting a ***label***'s ***for*** attribute to the ***id*** attribute of the related form field.

```
<h2>Text fields</h2>
<form id="text-fields">
  <label for="text-fields-username">Username:</label>
  <input id="text-fields-username" placeholder="jdoe"/><br/>
  <label for="text-fields-password">Password:</label>
  <input type="password" id="text-fields-password" value="123@#$asd"/><br/>
  <label for="text-fields-first-name">First name:</label>
  <input type="text" id="text-fields-first-name" title="John"/><br/>
  <label for="text-fields-last-name">Last name:</label>
  <input type="text" id="text-fields-last-name" placeholder="Doe"
```



Text fields

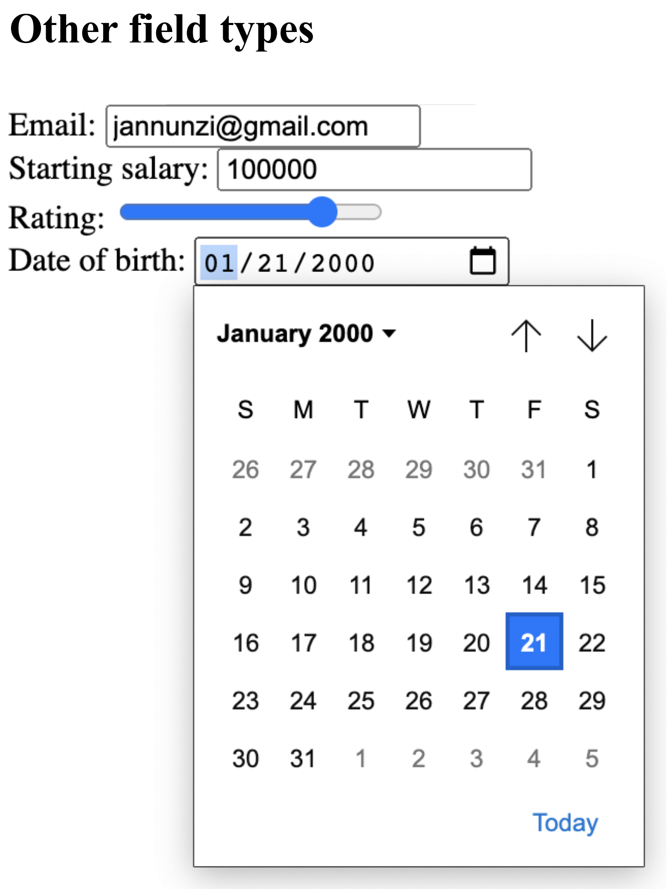
Username: jdoe
Password: •••••••••
First name:
Last name: Wonderland

```
        value="Wonderland"/>
    <!-- copy rest of form elements here -->
</form>
```

Date, email, number, and range fields

The input tag's ***type*** attribute has several other possible values: ***date***, ***email***, ***number***, and ***range***. They can be used to enter text information with a specific format. To practice these other formats add the following example under the last input field you worked on earlier, but inside the ***form*** tag. The fields should look as shown below on the right.

```
<h2>Other field types</h2>
<label for="text-fields-email">
        Email:
</label>
<input type="email"
        placeholder="jdoe@somewhere.com"
        id="text-fields-email"/><br/>
<label for="text-fields-salary-start">
        Starting salary:
</label>
<input type="number"
        id="text-fields-salary-start"
        placeholder="1000"
        value="100000"/><br/>
<label for="text-fields-rating">
        Rating:
</label>
<input type="range" id="text-fields-rating"
        placeholder="Doe"
        max="5"
        value="4"/><br/>
<label for="text-fields-dob">
        Date of birth:
</label>
<input type="date"
        id="text-fields-dob"
        value="2000-01-21"/><br/>
```



Other field types

Email: jannunzi@gmail.com

Starting salary: 100000

Rating:

Date of birth: 01/21/2000

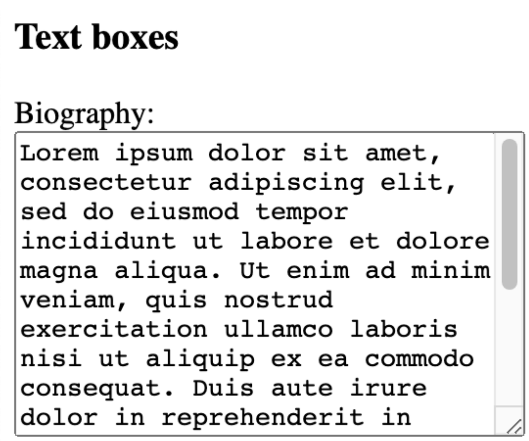
Text boxes

The ***texarea*** tag is useful for entering long form text such as someone's biography data, or a blog post.

```
<textarea cols="20"                          <!--    use texarea tag for long form text
        rows="25"                                     configure its width and height with attributes
        placeholder="Biography"                       cols and rows. Use placeholder and tooltip to give
        title="tooltip">Some text</textarea>          hints. Note default value is in tag's body -->
```

To practice using the ***textarea*** tag, add the following example to the end of index.html. It creates a ***textarea*** useful for entering your biography. You can get a sample of the dummy text at https://www.lipsum.com/.

```
<h2>Text boxes</h2>
<form id="textarea">
   <label>Biography:</label><br/>
   <textarea cols="30" rows="10">Lorem ipsum dolor sit amet,
consectetur adipiscing elit, sed do eiusmod tempor incididunt ut
labore et dolore magna aliqua. Ut enim ad minim veniam, quis
nostrud exercitation ullamco laboris nisi ut aliquip ex ea
commodo consequat. Duis aute irure dolor in reprehenderit in
voluptate velit esse cillum dolore eu fugiat nulla pariatur.
Excepteur sint occaecat cupidatat non proident, sunt in culpa
qui officia deserunt mollit anim id est laborum.</textarea>
</form>
```



Text boxes

Biography:

```
Lorem ipsum dolor sit amet,
consectetur adipiscing elit,
sed do eiusmod tempor
incididunt ut labore et dolore
magna aliqua. Ut enim ad minim
veniam, quis nostrud
exercitation ullamco laboris
nisi ut aliquip ex ea commodo
consequat. Duis aute irure
dolor in reprehenderit in
```

Buttons

Buttons allow invoking actions executed by the browser. To practice creating buttons, copy the code below at the end of ***index.html***.

```
<h3>Buttons</h3>
<button>Click me!</button>
```

Dropdowns

Dropdowns are useful for selecting one or more options from a list of possible values. The default version displays a set of values from which you can choose a single value.

```
<select>                                                <!--    Wrap several option tags in a select tag.
        <option value="VAL1">Value 1</option>                   optionally provide option's value,
        <option value="VAL2" selected>Value 2</option>          otherwise the option's text is the value
        <option value="VAL3">Value 3</option>                   of the select element. Optionally use
</select>                                                       selected attribute to select default.   -->
```

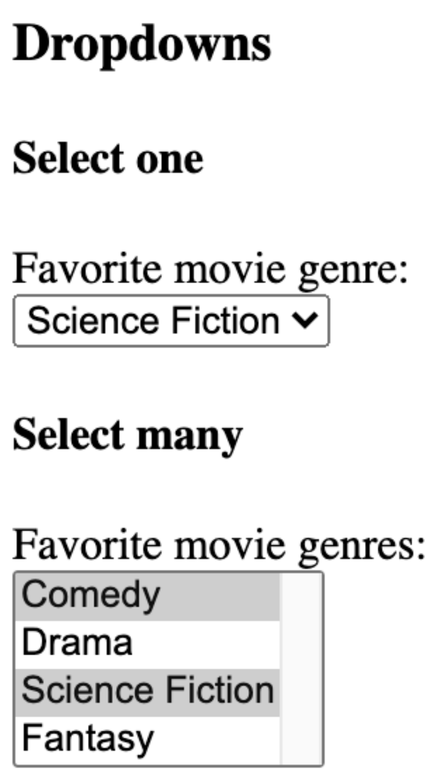
Adding the optional ***multiple*** attribute converts the dropdown into a list of options that can be selected.

```
<select multiple>                                       <!--    Alternatively use attribute multiple to
        <option value="VAL1" selected>Value 1</option>          allow selecting more than one option.
        <option value="VAL2">Value 2</option>                   Use ctrl+click to select more than one
        <option value="VAL3" selected>Value 3</option>          option -->
</select>
```

To practice using the ***select*** tag, add the following example to the end of ***index.html***. It creates a dropdown and a list of options.

```
<h2>Dropdowns</h2>
<h3>Select one</h3>
<label for="select-one-genre">
        Favorite movie genre:
</label><br/>
<select id="select-one-genre">
  <option value="COMEDY">Comedy</option>
  <option value="DRAMA">Drama</option>
  <option selected value="SCIFI">Science Fiction</option>
  <option value="FANTASY">Fantasy</option>
</select>

<h3>Select many</h3>
<label for="select-many-genre">
        Favorite movie genres:
</label><br/>
<select id="select-many-genre" multiple>
  <option selected value="COMEDY">Comedy</option>
  <option value="DRAMA">Drama</option>
  <option selected value="SCIFI">Science Fiction</option>
  <option value="FANTASY">Fantasy</option>
</select>
```



Dropdowns

Select one

Favorite movie genre:

Select many

Favorite movie genres:

File upload button

Use the file type for the input tag to choose a file for upload. We won't be able to upload just yet until later in the course, but for now let's practice adding a file upload tag as shown below. Clicking the button pops up a file choose where you can navigate to the file you want to upload. To practice using the file selector, copy the code below to the end of ***index.html***. We'll learn how to upload files later in the course.

```
<h2>File upload</h2>
<input type="file"/>
```

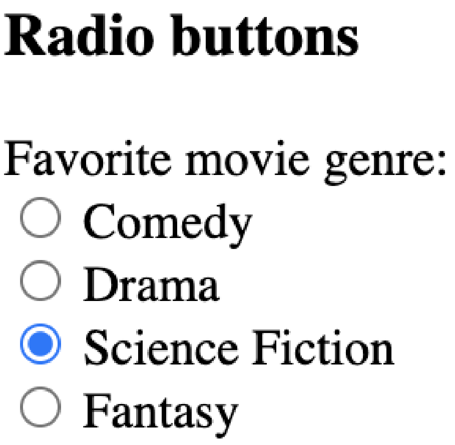
Radio buttons

Radio buttons allow selecting a single choice from multiple alternative options

```
<input  type="radio"              <!--    use the input tag's checkbox type to declare a checkbox
        name="NAME1"                      give the checkbox a value      -->
        value="OPTION1"/>
<input  type="radio" checked
        name="NAME1"
        value="OPTION2"/>
```

To practice using radio buttons, add the following example at the end of ***index.html***.

```
<h2>Radio buttons</h2>
<label>Favorite movie genre:</label><br/>
<input type="radio" value="COMEDY"
        name="radio-genre" id="radio-comedy"/>
<label for="radio-comedy">Comedy</label><br/>
<input type="radio" value="DRAMA"
        name="radio-genre" id="radio-drama"/>
<label for="radio-drama">Drama</label><br/>
<input type="radio" value="SCIFI"
        name="radio-genre" id="radio-scifi" checked/>
<label for="radio-scifi">Science Fiction</label><br/>
<input type="radio" value="FANTASY"
        name="radio-genre" id="radio-fantasy"/>
<label for="radio-fantasy">Fantasy</label>
```



Radio buttons

Favorite movie genre:
○ Comedy
○ Drama
● Science Fiction
○ Fantasy

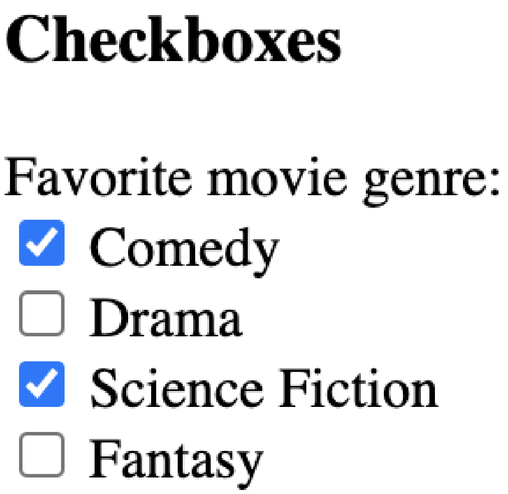
Checkboxes

Checkboxes allow selecting multiple choices

```
<input  type="checkbox"         <!--     use the input tag's checkbox type to declare a checkbox
        name="NAME2"                     give the checkbox a value      -->
        checked
        value="OPTION1"/>
<input  type="checkbox"
        name="NAME2"
        value="OPTION2"/>
<input  type="checkbox"
        checked
        name="NAME2"
        value="OPTION3"/>
```

To practice using checkboxes, add the following example to the end of ***index.html***. It creates a set of checkbox buttons to select all your favorite movie genres, which there might be more than one.

```
<h2>Checkboxes</h2>
<label>Favorite movie genre:</label>
<br/>
<input type="checkbox" value="COMEDY"
        name="check-genre" id="chkbox-comedy" checked/>
<label for="chkbox-comedy">Comedy</label>   <br/>
<input type="checkbox" value="DRAMA"
        name="check-genre" id="chkbox-drama"/>
<label for="chkbox-drama">Drama</label>   <br/>
<input type="checkbox" value="SCIFI"
        name="check-genre" id="chkbox-scifi" checked/>
<label for="chkbox-scifi">Science Fiction</label>   <br/>
<input type="checkbox" value="FANTASY"
        name="check-genre" id="chkbox-fantasy"/>
<label for="chkbox-fantasy">Fantasy</label>
```



Checkboxes

Favorite movie genre:
☑ Comedy
☐ Drama
☑ Science Fiction
☐ Fantasy

Anchor Tag

The anchor tag allows navigating to other websites or other pages within the same website.

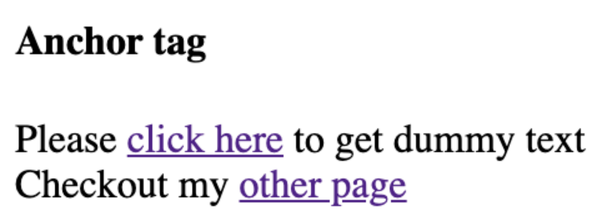
```
<a href="aa.com">         <!--     Use the href attribute to refer to the location of the website
American Airlines</a>              or other page in the same website. Click on the body text to navigate -->
```

To practice using anchor tags, add the following example at the end of ***index.html***. It creates two hyperlinks. One navigates to ***lipsum.com***, a website that contains dummy text, and the other link navigates to another document located in the same website. Create the ***other-page.html*** document in the same directory as ***index.html*** and fill it with some dummy text. Confirm navigation works.

```
<h2>Anchor tag</h2>

Please
<a href="https://www.lipsum.com">click here</a>
to get dummy text<br/>

Checkout my <a href="other-page.html">other page</a>
```



Anchor tag

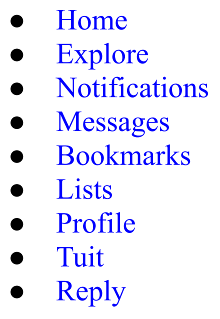
Please click here to get dummy text
Checkout my other page

Tuiter

Now that you've had a chance to practice various aspects of HTML, we're going to use them to build ***Tuiter***, a social network Website. We'll start with some simple versions of the more common screens and we'll improve on it over several assignments. Do your work under a new ***public/tuiter*** directory.

Navigation screen

In ***public/tuiter/navigation.html*** create an unordered list of hyperlinks referencing other HTML documents as shown below. We will create some of these documents here and others in later assignments.

Label	HREF	Browser
Home	home.html	 Home Explore Notifications Messages Bookmarks Lists Profile Tuit Reply
Explore	explore.html	
Notifications	notifications.html	
Messages	messages.html	
Bookmarks	bookmarks.html	
Lists	lists.html	
Profile	profile.html	
Tuit	tuit.html	
Reply	reply.html	

To help ***Teaching Assistants*** navigate your Website, create ***public/labs/a2/index.html*** as shown below. Add links to the lab exercises for this assignment and add links to ***index.html*** as you work through other lab exercises.

public/labs/a2/index.html

```
<a href="../../index.html">Back</a>
<h1>Assignment 2</h1>
<ul>
   <li><a href="html/index.html">HTML Exercises</a></li>
   <!-- add more links to the rest of the exercises later in this assignment -->
</ul>
```

Also replace the content of the ***<body>*** tag in ***public/index.html*** with the following content. Run the file by right clicking it and then selecting ***Run index.html***. Confirm that you can navigate to the lab exercises and the Tuiter navigation screen.

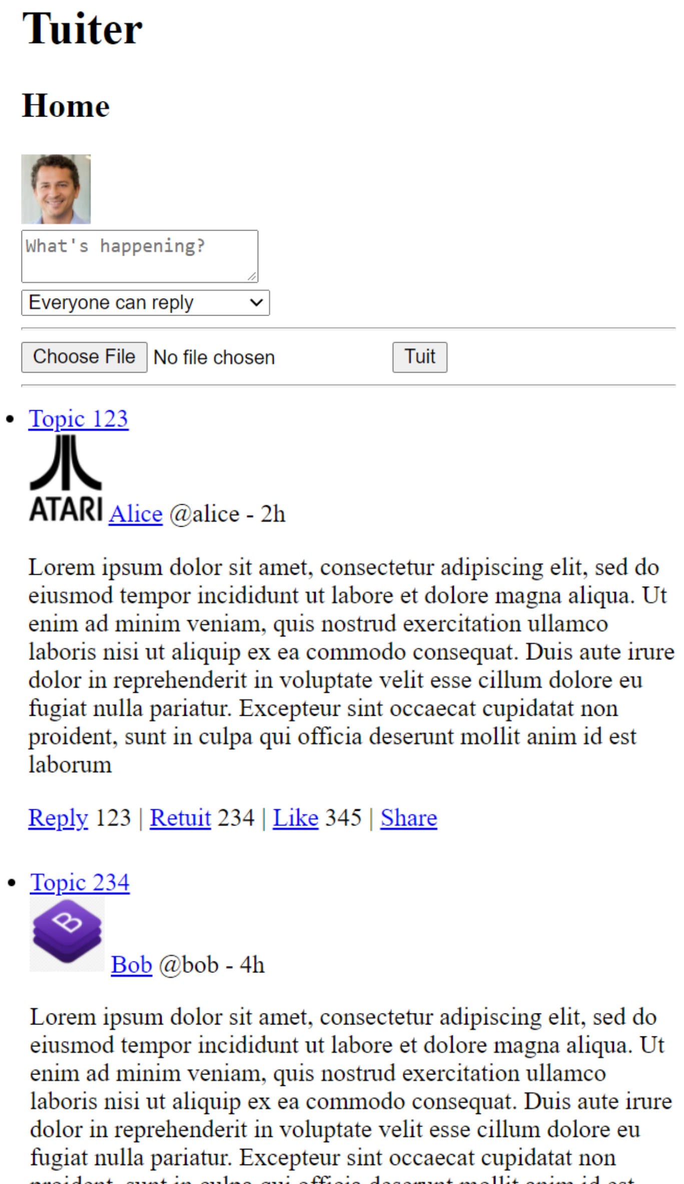
public/index.html

```
<h1>Web Dev Assignments</h1>
<ul>
   <li><a href="labs/a2/index.html">Assignment 2 Labs</a></li>
   <li><a href="tuiter/navigation.html">Tuiter</a></li>
</ul>
```

Home screen

In ***public/tuiter/home.html*** create the Home screen for users to view tuit posts and create new posts as shown down below. Like most screens, add a ***Back*** link to navigate back to the previous screen, ***navigation.html*** in this case. Below the link add an h1 ***Tuiter*** header and an h2 ***Home*** header below that. Below the headers add a 48 by 48 pixel ***image*** representing the owner of the account. Use an image of your choice. Add a ***textarea*** element for users to post new tuits. The textarea should have a ***placeholder*** that says "***What's happening?***". Add a ***dropdown*** with options "***Everyone can reply***", "***People you follow***", and "***Only people you mention***", with values "***EVERYONE***", "***FOLLOWING***", and "***MENTIONED***" respectively. ***Default*** value should be "***EVERYONE***". Add a ***button*** to upload an image and another button labeled "***Tuit***" to create a new tuit. The page should look as shown on the right.

Use an unordered list ***ul*** to create a couple of tuits as line item elements ***li*** shown here on the right as bullet points. As the header of the tuit, add a hyperlink referencing ***topic.html*** labeled ***Topic 123***. Add a 48 by 48 pixel image, a link referencing ***profile.html*** labeled ***Alice***, add a handle ***@alice***, and a time stamp of ***2 hours*** as



Tuiter

Home

What's happening?

Everyone can reply ⌄

Choose File No file chosen Tuit

- Topic 123

  Alice @alice - 2h

  Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

  Reply 123 | Retuit 234 | Like 345 | Share

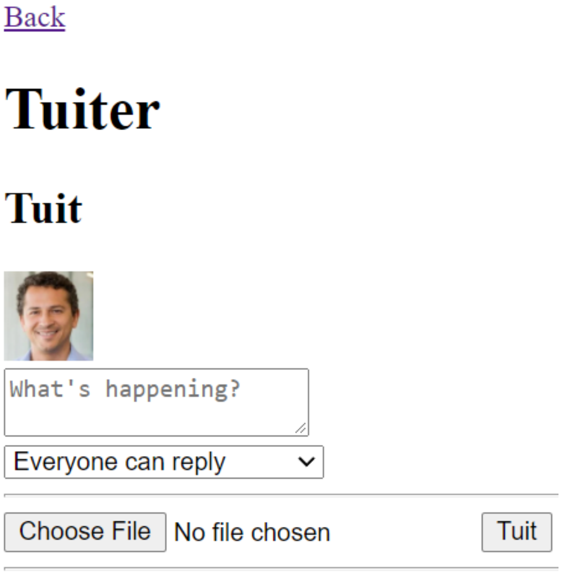
- Topic 234

  Bob @bob - 4h

  Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est shown on the right. Below the tuit's header add a paragraph with dummy content as shown. Below the tuit, add ***Reply***, ***Retuit***, ***Like***, and ***Share*** links linked to ***reply.html***, ***retuit.html***, ***like.html***, and ***share.html*** respectively. Include count values ***123***, ***234***, and ***345*** as shown. Instead of ***Alice*** and ***@alice***, use your ***first name***.

Tuit screen

Create a screen for posting new tuits. In a document in ***public/tuiter/tuit.html***, create a screen with the content shown here on the right. Add a ***Back*** link to navigate back to ***navigation.html***. Add an h1 ***Tuiter*** header at the top of the document. Add a smaller h2 ***Tuit*** header under that. Below the headers, add a 48x48 pixel ***image*** representing the owner of the account. Add a ***textarea*** element for users to post new tuits. The textarea should have a ***placeholder*** that says ***What's happening?***. Add a dropdown with options ***Everyone can reply***, ***People you follow***, and ***Only people you mention***, with values ***EVERYONE***, ***FOLLOWING***, and ***MENTIONED*** respectively. Default value should be ***EVERYONE***. Add a button to upload an image. Add a button labeled ***Tuit*** to create the new tuit. The page should look as shown on the right.



Back

Tuiter

Tuit

What's happening?

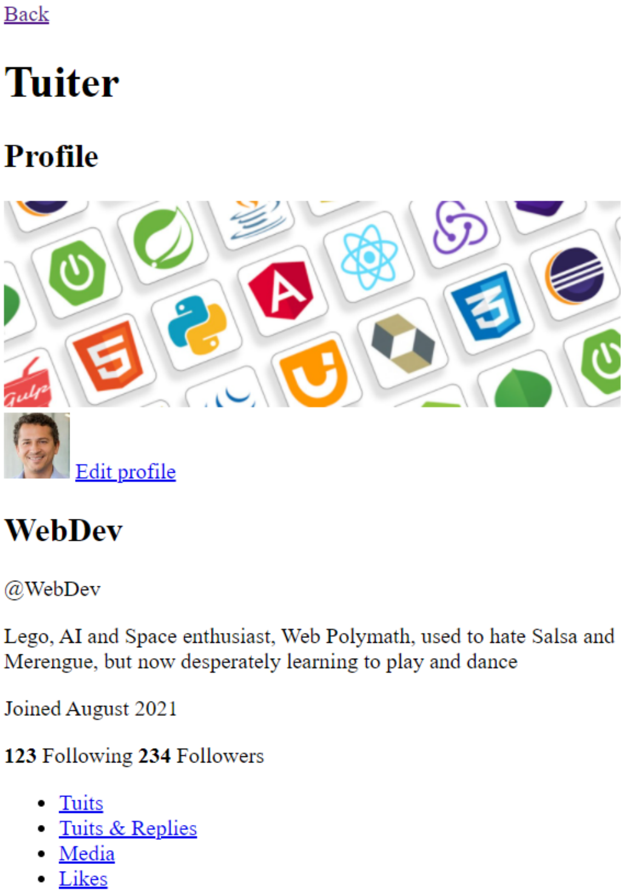
Everyone can reply

Choose File No file chosen

Tuit

Profile screen

In ***public/tuiter/profile.html***, create a screen that renders a user's profile as shown here on the right. Like most screens, add a ***Back*** link to navigate to the previous screen, ***navigation.html*** in this case. Add an h1 ***Tuiter*** header below the ***Back*** link, and an h2 ***Profile*** header below the Tuiter header. Below the Profile header, add a banner ***image*** that is 200 pixels high and 100% in width. Use an image of your choice. Below the banner image, add an avatar image representing the owner of the account of 48 x 48 pixels. Use an image of your choice. Next to the avatar image add an ***Edit profile*** link referencing ***edit-profile.html***. Below the avatar image, add the name of the user, ***WebDev***, using an h2 header. Add the user's handle, ***@WebDev***, below the user's name. Below the handle, add a paragraph describing the user's bio. Below the bio add another paragraph with the date the user joined. Below the join date, add another paragraph with the numbers of users being followed and following. Bold the numbers. Finally, at the bottom of the profile screen, add links ***Tuits***, ***Tuits & replies***, ***Media***, and ***Likes*** referencing ***tuits.html***, ***tuits-and-replies.html***, ***media.html***, and ***likes.html***.



Back

Tuiter

Profile

Edit profile

WebDev

@WebDev

Lego, AI and Space enthusiast, Web Polymath, used to hate Salsa and Merengue, but now desperately learning to play and dance

Joined August 2021

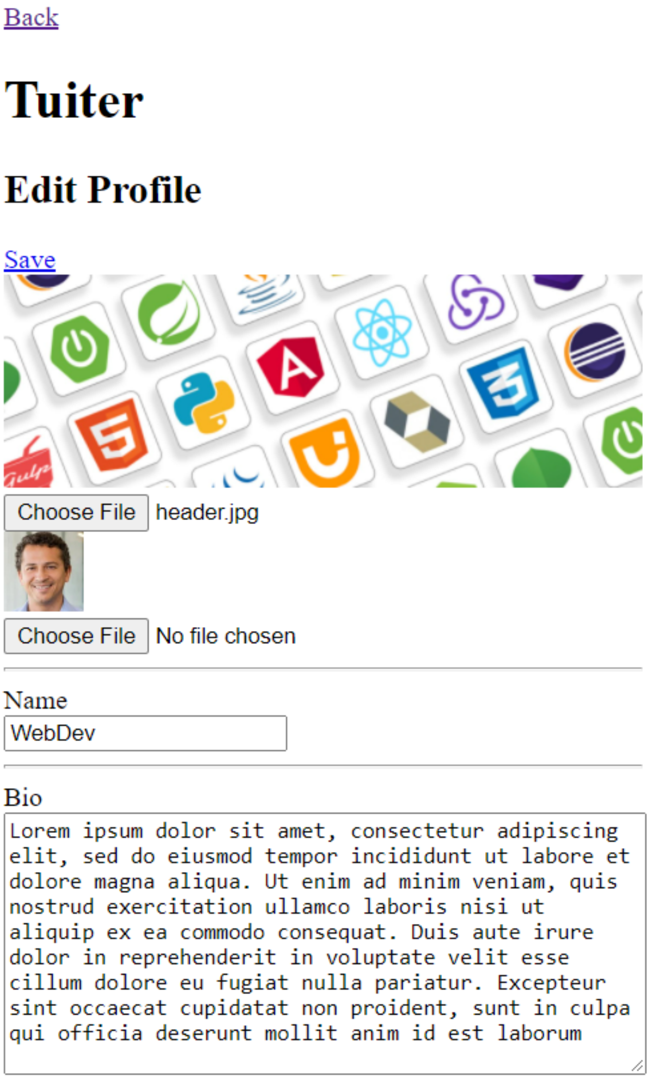
123 Following **234** Followers

- Tuits
- Tuits & Replies
- Media
- Likes

Edit profile screen

In ***public/tuiter/edit-profile.html***, add a ***Back*** button at the top navigating to the previous screen, e.g., ***profile.html***. Below the Back link, add an h1 ***Tuiter*** header and an h2 ***Edit Profile*** header below that. Below the headers add a ***Save*** link referencing ***profile.html***. Then add a 200 pixel high by 100% wide ***banner image*** below the Save link. Use an image of your choice. Below the image add a ***file input field*** that could be used in the future to update the banner image. Then add a 48 x 48 pixel ***avatar image***. Use an image of your choice. Below the avatar image add a ***file input field*** that could be used in the future to update the avatar image. Add a ***ruler*** tag to separate the content vertically. Below the ruler add a ***Name*** label associated to an ***input*** field that in the future could be used to edit the name of the user. The field should have ***WebDev*** as the devault value and ***Name*** as the placeholder. If you click on the label, the input field should get focus. Add a ***ruler*** tag to separate the content vertically. Below the ruler tag, add a ***Bio*** label associated to a textarea that in the future could be used to edit the user's bio information. The default bio value should be the dummy ***Lorem ipsum*** text shown. Find an example text at lipsum.com. The textarea should be 50 columns wide by 10 rows high.

Confirm you can navigate from ***navigation.html*** to ***home.html***, and ***profile.html*** and then back to ***navigation.html***. Also confirm you can navigate from profile.html to ***edit-profile.html*** and back again.



Back

Tuiter

Edit Profile

Save

header.jpg

No file chosen

Name

WebDev

Bio

```
Lorem ipsum dolor sit amet, consectetur adipiscing
elit, sed do eiusmod tempor incididunt ut labore et
dolore magna aliqua. Ut enim ad minim veniam, quis
nostrud exercitation ullamco laboris nisi ut
aliquip ex ea commodo consequat. Duis aute irure
dolor in reprehenderit in voluptate velit esse
cillum dolore eu fugiat nulla pariatur. Excepteur
sint occaecat cupidatat non proident, sunt in culpa
qui officia deserunt mollit anim id est laborum
```

Challenge

The following are a set of practice sections exploring some additional features of HTML. They are optional for undergraduate students, but are required for graduate students. For undergraduate students, the exercises will not be graded, but they are nevertheless encouraged to go through the exercises.

Iframes

Iframes allow sharing entire webpages within your own Web page. You can define a rectangle where the browser loads another Web page and renders it into that rectangle. Complete the examples below in a new

HTML document in ***public/labs/a2/iframe/index.html***. Add a link to this new exercise in ***public/labs/a2/index.html***.

Embedding other Web sites

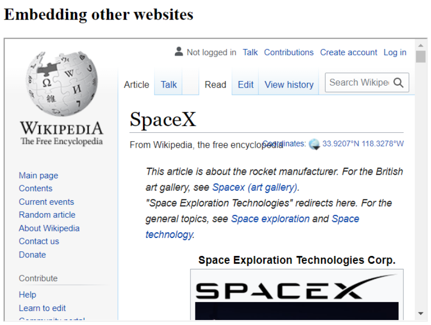
Embedding another Web site into your own is very easy using the ***iframe*** tag. Just configure the size of the rectangle you want to dedicate to the other Web site and the URL where the other Web site lives. Here's the syntax of the ***iframe*** tag.

```
<iframe width="600"                                    <!--    width and height attributes configure the size
      height="400"                                             of the rectangle we want to dedicate to the
      src="https://en.wikipedia.org/wiki/SpaceX">              other Web site. the src attribute points to the
</iframe>                                                      location of the other Web site -->
```

To practice embedding other Web sites, on your browser, navigate to a Web site you want to embed, copy the URL, use the URL in the ***src*** attribute of an iframe tag as shown below. Copy the code below into ***index.html***. Feel free to replace the URL we used with one of your own. Keep it PG-13.

public/labs/a2/iframe/index.html

```
<h2>Embedding other websites</h2>
<iframe width="600"
      height="400"
      src="https://en.wikipedia.org/wiki/SpaceX">
</iframe>
```



Embedding other websites

SpaceX

From Wikipedia, the free encyclopedia Coordinates: 33.9207°N 118.3278°W

This article is about the rocket manufacturer. For the British art gallery, see Spacex (art gallery).
"Space Exploration Technologies" redirects here. For the general topics, see Space exploration and Space technology.

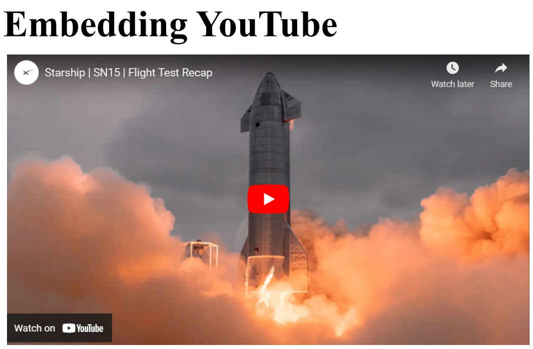
Space Exploration Technologies Corp.

Embedding a YouTube video

Other applications such as YouTube and Google slides take advantage of iframes to allow content creators to share their content. Let's take a look at how to share a YouTube video on your Web site. Find a video on YouTube you want to embed, click on ***Share***, click ***Embed***, copy the code under Embes Video, paste the iframe code into your ***index.html***. Feel free to copy the code below.

public/labs/a2/iframe/index.html

```
<h2>Embedding YouTube</h2>
<iframe width="560"
    height="315"
src="https://www.youtube.com/embed/7CZTLogIn34"
    title="YouTube video player"
    frameborder="0"
    allow="accelerometer; autoplay; clipboard-write;
encrypted-media; gyroscope; picture-in-picture"
    allowfullscreen>
</iframe>
```



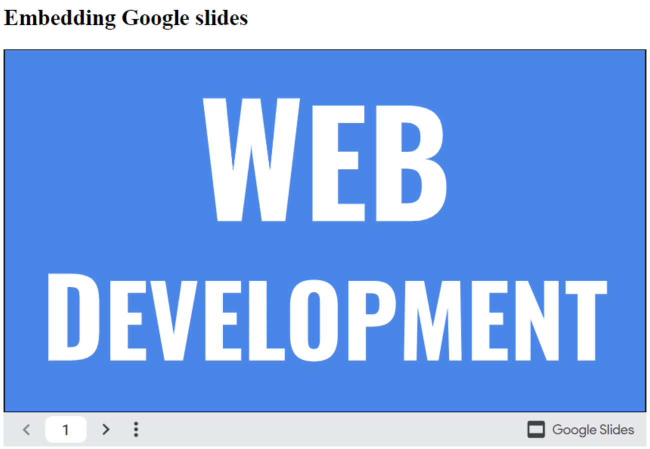
Embedding YouTube

Embedding Google slides

Now let's practice embedding slides from Google slides. Find a set of slides you want to share, then click on the menu ***File***, ***Publish*** to the Web, ***Embed***, copy the code snippet with iframe, paste the iframe code to your ***index.html***. Feel free to copy the code below.

public/labs/a2/iframe/index.html

```
<h2>Embedding Google slides</h2>
<iframe
src="https://docs.google.com/presentation/d/e/2PACX
-1vQoIYM2gXzRLKt4q5D2-ZahG7zvVcAipkRvj5k9GVLiG
Ds8NPlLpO3Q08cPUKQbqhL6g8aZp1ZJSwrw/embed?
start=false&loop=false&delayms=3000"
	frameborder="0"
	width="680"
	height="418"
	allowfullscreen="true"
	mozallowfullscreen="true"
	webkitallowfullscreen="true">
</iframe>
```



Embedding Google slides

Scalable vector graphics (SVG)

The ***svg*** tag allows declaring a rectangle where we can draw shapes on the screen. The svg tag can contain shape tags such as ***rect***, ***circle*** and ***polygon*** to draw anything we want. Here's an example syntax of the svg, rect, and circle tags.

```
<svg    width="300"                           <!--   svg declares a rectangular area where we can draw
        height="100">                                width and height attributes configure the size of the area
  <rect width="300"                                  rect tag declares a rectangular area to draw a rectangle
        height="100"                                 width and height attributes declare size of rectangle
        style="fill:rgb(0,0,255);                    style attribute declares border and fill colors and border
        stroke:rgb(0,0,0);                           width
        stroke-width:10;
        " />
  <circle cx="50" cy="50"                            circle tag declares a rectangular area to draw a circle
        r="40"                                       cx and cy attributes declares the center position
        stroke="green"                               r attribute declares the radius of the circle
        stroke-width="4"                             stroke declares the color to draw with
        fill="yellow" />                             stroke-width declares the size of the border
</svg>                                               fill declares color to fill the circle -->
```

To practice drawing shapes on a browser window, create a new HTML document in ***public/labs/a2/svg/index.html***. A a link to the new exercise in ***public/labs/a2/index.html***.

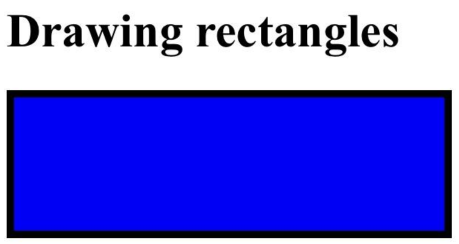
Drawing rectangles

Let's first practice drawing a rectangle. Copy the code below into the new ***index.html*** document and confirm it renders as shown in the right. Feel free to change the border and fill colors.

public/labs/a2/svg/index.html

```
<h2>Drawing rectangles</h2>

<svg width="300" height="100">
  <rect width="300" height="100"
    style="fill:rgb(0,0,255); stroke-width:10;
        stroke:rgb(0,0,0)" />
</svg>
```



Drawing rectangles

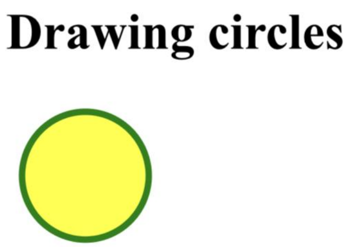
Drawing circles

Let's now practice drawing a circle. Copy the code below into the new ***index.html*** document and confirm it renders as shown in the right. Feel free to change the border and fill colors.

public/labs/a2/svg/index.html

```
<h2>Drawing circles</h2>

<svg width="100" height="100">
  <circle cx="50" cy="50" r="40"
    stroke="green" stroke-width="4" fill="yellow" />
</svg>
```



Drawing circles

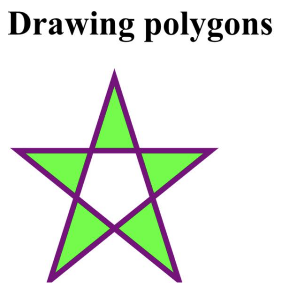
Drawing polygons

The ***polygon*** tag allows drawing any shape. Let's now practice drawing a polygon. Copy the code below into the new ***index.html*** document and confirm it renders as shown in the right. Feel free to change the border and fill colors.

public/labs/a2/svg/index.html

```
<h2>Drawing polygons</h2>

<svg width="300" height="200">
  <polygon points="100,10 40,198 190,78 10,78 160,198"
    style="fill:lime; stroke:purple; stroke-width:5; fill-rule:evenodd;" />
</svg>
```



Drawing polygons

Anchors

We used anchor tags earlier in this assignment to link one Web page to another. The ***href*** attribute in anchor tags allow referring other documents which we navigate to when we click on the anchor. We can also reference sections in the same document so we can navigate to different places in the document. To do this we use the

name attribute to declare places in the document we want to navigate to. Here's the syntax for implementing this:

```
<a name="table-of-content"></a>          <!--    declaring a location on the page called table-of-content
<a href="#section-a>Go to section A</a>          link to go to section-a
<a href="#section-b>Go to section B</a>          link to go to section-b

<a name="section-a"></a>                         declaring location on the page called section-a
<h1>Section A</a>                                actual content
<a href="#table-of-content>Up</a>                link to go to table-of-content

<a name="section-b"></a>                         declaring location on the page called section-b
<h1>Section B</a>                                actual content
<a href="#table-of-content>Up</a>                link to go to table-of-content -->
```

Table of content anchors

To practice using anchors, let's add a table of content to the iframes exercise page. Add the code highlighted in yellow bellow to the top of the ***index.html*** you created to practice iframes.

public/labs/a2/iframe/index.html

```
<a name="top"></a>
<h2>Table of content</h2>
<ul>
   <li><a href="#embedding-website">
         Embedding other websites</a></li>
   <li><a href="#embedding-youtube">
         Embedding YouTube videos</a></li>
   <li><a href="#embedding-slides">
         Embedding Google slide</a></li>
</ul>

<a name="embedding-website"></a>
<h2>Embedding other websites</h2>
<a href="#top">Top</a><br/>

<a name="embedding-youtube"></a>
<h2>Embedding YouTube</h2>
<a href="#top">Top</a><br/>

<a name="embedding-slides"></a>
<h2>Embedding Google slides</h2>
<a href="#top">Top</a><br/>
```

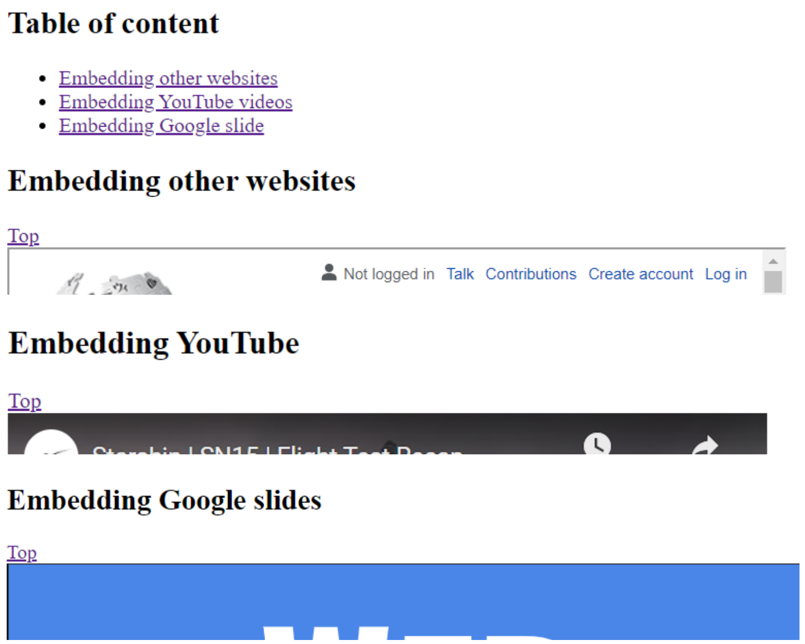


Table of content

- Embedding other websites
- Embedding YouTube videos
- Embedding Google slide

Embedding other websites

Top

Embedding YouTube

Top

Embedding Google slides

Top

This creates hyperlinks to anchors ***#embedding-website***, ***#embedding-youtube*** and ***#embedding-slides***. Now add the code highlighted red above each of the sections ***Embedding other websites***, ***Embedding YouTube***, and ***Embedding Google slides***. Save the content and refresh the iframe practice ***index.html*** Web page. Confirm that clicking the links in the table of content navigates down to the sections below. Now add the code highlighted blue below each of the same sections as before. Save and refresh the page and confirm that the ***Top*** hyperlinks navigate back to the top of the screen. Add a similar table of content to the SVG examples.

Deliverables

Commit all your changes, push to your GitHub repository, and deploy to the remote server. Submit the link to your GitHub repository and the remote server. Add, commit and push all your code to your remote repository as follows:

```
git  add  .
git  commit  -am  "a2 html sp22"
git  push
```