CP 160 - Web Programming and Design

Lab 3: HTML Basics 2

Hands-On Practice 1: Structural Elements

HTML5 introduces a number of semantic structural elements that can be used along with the generic div element to configure specific areas on a web page. **Figure 1** shows a diagram of a page (called a **wireframe**) that indicates how the structure of a web page could be configured with the header, nav, main, div, and footer elements.

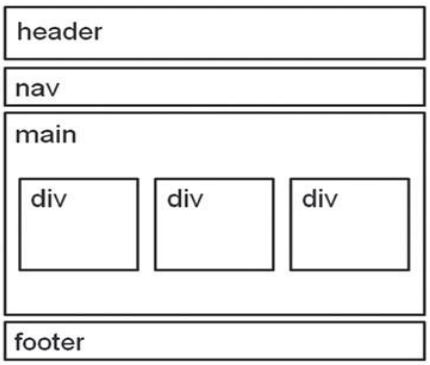

Figure 1 Structural elements.

In this Hands-On Practice, you will use structural elements as you create the Trillium Media Design home page, shown in **Figure 2**. Launch a text editor, and open the **template.html** file. Edit the code as follows:

1. Modify the title of the web page by changing the text between the `<title>` and `</title>` tags to Trillium Media Design.

2. Position your cursor in the body section and code the header element with the text, "Trillium Media Design" in an h1 element:

```
<header>
  <h1>Trillium Media Design</h1>
```

```
</header>
```

3. Code a nav element to contain text that will indicate the main navigation for the website. Configure bold text (use the b element) and use the ` ` special character to add extra blank space:

```
<nav>
  <b>Home   Services   Contact</b>
</nav>
```

4. Code the content within a main element that contains the h2 and paragraph elements:

```
<main>
  <h2>New Media and Web Design</h2>
  <p>Trillium Media Design offers a comprehensive range of services to take your company&rsquo;s Web presence to the next level.</p>
  <h2>Meeting Your Business Needs</h2>
  <p>Our expert designers will listen to you as they create a website that helps to promote and grow your business.</p>
</main>
```

5. Configure the footer element to contain a copyright notice displayed in small font size (use the small element) and italic font (use the i element). Be careful to properly nest the elements as shown here:

```
<footer>
  <small><i>Copyright &copy; 2020 Your Name Here</i></small>
</footer>
```

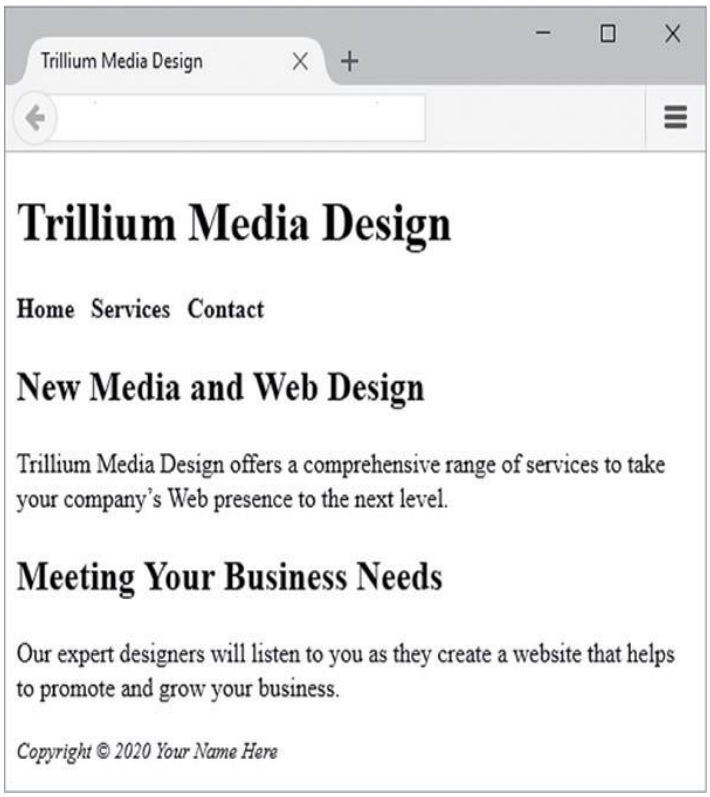

Figure 2 Trillium home page.

Save your page as structure.html. Test your page in a browser. It should look similar to **Figure 2**.

Hands-On Practice 2: Anchor Element

Use the anchor element to specify a **hyperlink,** often referred to as a *link*, to another web page or file that you want to display. Each anchor element begins with an **<a>** tag and ends with an **** tag.

The opening and closing anchor tags surround the text that the user can click to perform the hyperlink. Use the **href attribute** to configure the hyperlink reference, which identifies the name and location of the file to access.

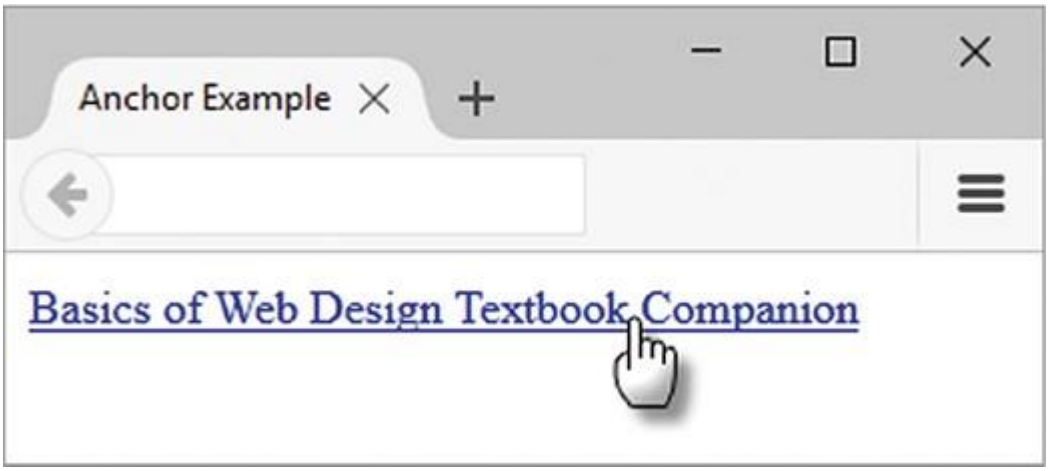

Figure 3 Sample hyperlink.

To create the web page shown in **Figure 3**, launch a text editor and open the **template.html** file. Modify the title element and add an anchor tag to the body section as indicated by the following highlighted code:

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Anchor Example</title>
<meta charset="utf-8">
</head>
<body>
<a href="http://webdevbasics.net">Basics of Web Design Textbook
Companion</a>
</body>
</html>
```

Save the document as anchor.html on your hard drive or flash drive. Launch a browser and test your page. It should look similar to the page shown in **Figure 3.**

Hands-On Practice 3: Content Pages

In this practice, you will create a website with multiple content pages.

Figure 4 displays the site map for your new website—a home page (index.html) with two content pages: services page (services.html) and contact page (contact.html).

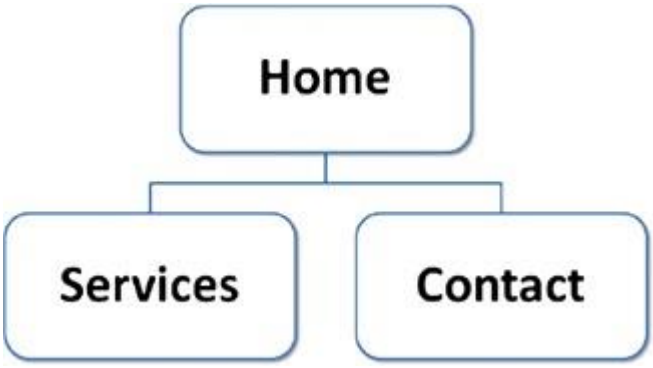

Figure 4 Site map.

1. **Create a Folder.**

   Use your operating system to create a new folder named **mypractice** for your new website.

2. **Create the Home Page.**

   Use the Trillium Media Design web page (**Figure 2**) from **Hands-On Practice 1** as a starting point for your new home page (shown in **Figure 5**). Copy the file **structure.html** into your new folder. Change the file name of structure.html to index.html. It's common practice to use the file name index.html for the home page of a website.

   Launch a text editor and open the index.html file.

   a. The navigation hyperlinks will be located within the nav element. You will edit the code within the nav element to configure three hyperlinks:

      - The text "Home" will hyperlink to index.html
      - The text "Services" will hyperlink to services.html
      - The text "Contact" will hyperlink to contact.html

   Modify the code within the nav element as given below:

```
<nav>
  <b>
```

```
    <a href="index.html">Home</a>  
    <a href="services.html">Services</a>  
    <a href="contact.html">Contact</a>
  </b>
</nav>
```

b. Save the index.html file in your folder. Test your page in a browser. It should look similar to **Figure 5**.

3. Create the Services Page.

It is common practice to create a new web page based on an existing page. You will use the index.html file as a starting point for the new Services page.

Open your index.html file in a text editor and save the file as services.html. Edit the code as indicated below:

a. Modify the title of the web page by changing the text between the `<title>` and `</title>` tags to "Trillium MediaDesign - Services". In order to create a consistent header, navigation, and footer for the web pages in this website, do not change the code within the header, nav, or footer elements.

b. Position your cursor in the body section and delete the code and text between the opening and closing main tags. Code the main page content (heading 2 and description list) for the Services page between the main tags as follows:

```
<h2> Our Services Meet Your Business Needs</h2>
  <dl>
    <dt><strong>Website Design</strong></dt>
      <dd>Whether your needs are large or small, Trillium can get you on the Web!</dd>
<dt><strong>E-Commerce Solutions</strong></dt>
 <dd>Trillium offers quick entry into the e-commerce marketplace.</dd>
<dt><strong>Search Engine Optimization</strong></dt>
     <dd>Most people find new sites using search engines. Trillium can get your website noticed.</dd>
</dl>
```

c. Save the services.html file in your mypractice folder. Test your page in a browser. It should look similar to **Figure 6**.

4. **Create the Contact Page.**

   Use the index.html file as a starting point for the Contact page. Launch a text editor, open index.html, and save the file as contact.html. Edit the code as indicated below:

   a. Modify the title of the web page by changing the text between the `<title>` and `</title>` tags to “Trillium Media Design - Contact”. In order to create a consistent header, navigation, and footer for the web pages in this website, do not change the code within the header, nav, or footer elements.

   b. Position your cursor in the body section and delete the code and text between the opening main tag and the closing main tag. Code the main page content for the Contact page between the main tags:

```
<h2>Contact Trillium Media Design Today</h2>
<ul>
  <li>E-mail: contact@trilliummediadesign.com</li >
      <li>Phone: 555-555-5555</li>
    </ul>
```

   c. Save the contact.html file in your mypractice folder. Test your page in a browser. It should look similar to **Figure 7**. Test your page by clicking each link. When you click the “Home” hyperlink, the index.html page should be displayed. When you click the “Services” hyperlink, the services.html page should be displayed. When you click the “Contact” hyperlink, the contact.html page should be displayed.

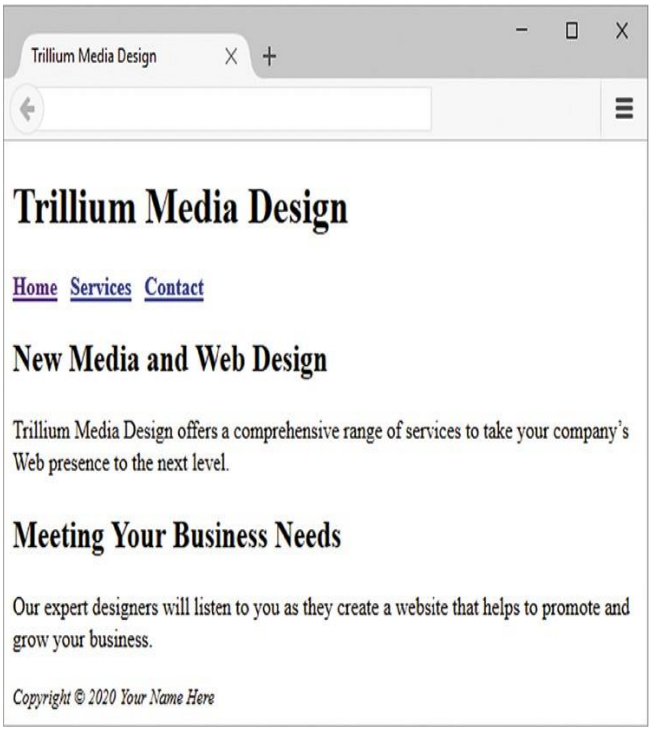

Figure 5 New index.html web page.

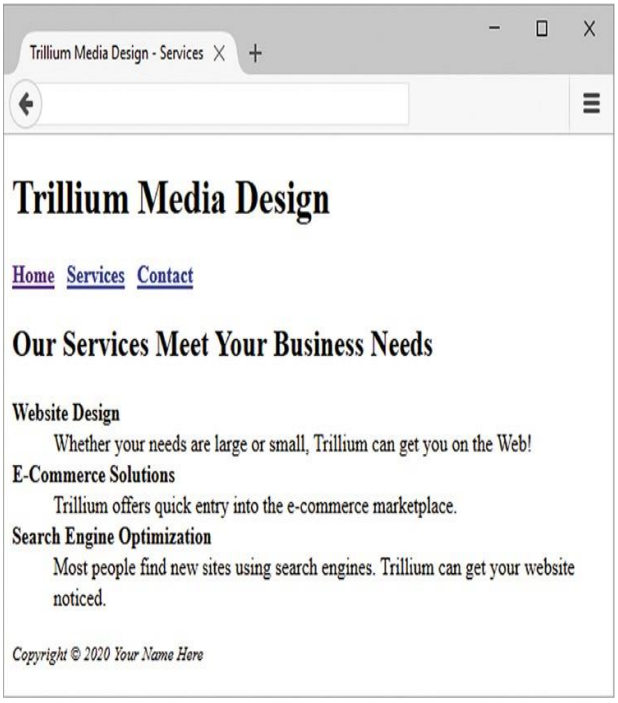

Figure 6 The services.html web page.

Figure 7 The contact.html web page.

Hands-On Practice 4: E-Mail Hyperlinks

The anchor tag can also be used to create e-mail hyperlinks. An email hyperlink will automatically launch the default mail program configured for the browser. It is similar to an external hyperlink with the following two exceptions:

- It uses `mailto:` instead of `http://`.
- It launches the default e-mail application for the visitor's browser with your e-mail address as the recipient.

For example, to create an e-mail hyperlink to the e-mail address **help@webdevbasics.net**, code the following:

```
<a href="mailto:help@webdevbasics.net">help@webdevbasics.net</a>
```

It is good practice to place the e-mail address both on the web page and within the anchor tag. Not everyone has an e-mail program configured with his or her browser. By placing the e-mail address in both places, you increase usability for all of your visitors.

In this Hands-On Practice, you will modify the Contact page (contact.html) of the website you created in **Hands-On Practice 3** and configure an e-mail link in the page content area. Launch a text editor and open the contact.html file from your mypractice folder.

Configure the e-mail address in the content area as an e-mail hyperlink as given below:

```
<li>E-mail: <a href
="mailto:contact@trilliummediadesign.com">contact@trilliummediades
ign.com</a>
</li>
```

Save and test the page in a browser. The browser display should look similar to the page shown in **Figure 8**.

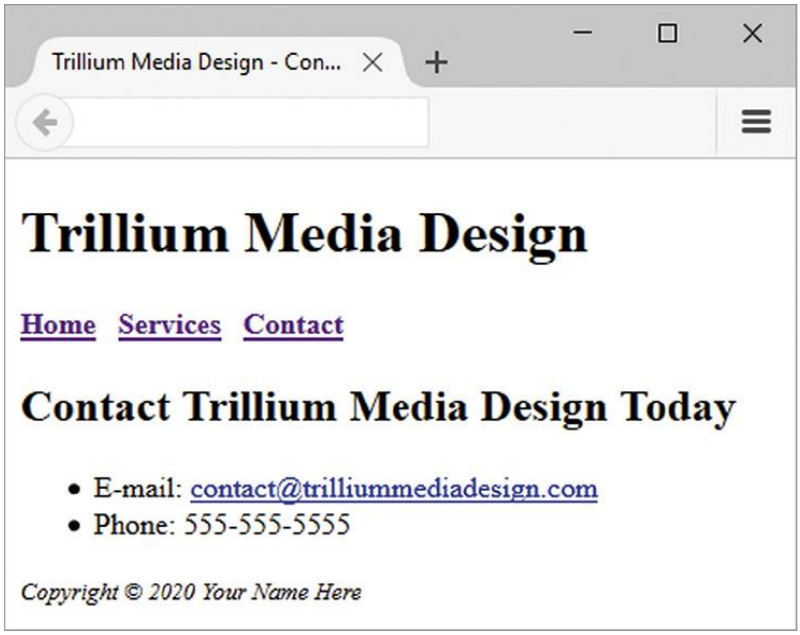

Figure 8 An e-mail hyperlink has been configured on the Contact page.

Hands-On Practice 5: HTML Syntax Validation

<if you have no time to complete this practice in the virtual lab online, you can complete it offline after the class time>

The W3C has a free Markup Validation Service available at **http:// validator.w3.org** that will check your code for syntax errors and validate your web pages. **HTML validation** provides you with quick self-assessment—you can prove that your code uses correct syntax. In the working world, HTML validation serves as a quality assurance tool. Invalid code may cause browsers to render the pages slower than otherwise.

In this Hands-On Practice, you will use the W3C Markup Validation Service to validate a web page file.

Launch a text editor and open the **design.html** file that you edited in Lab 2.

1. We will add an error to the design.html page. Delete the first closing `</strong>` tag. This modification should generate several error messages.
2. Next, attempt to validate the design.html file. Launch a browser and visit the W3C Markup Validation Service file upload page at **http:// validator.w3.org** and select the "Validate by File Upload" tab. Click the Browse button and select the chapter2/design.html file from your computer. Click the Check button to upload the file to the W3C site (**Figure 9**).
3. A results page will be displayed. Scroll down the page to view the errors, as shown in **Figure 10**.
4. Notice that the message indicates line 12, which is the first line after the missing closing `</strong>` tag. HTML error messages often point to a line that follows the error. The text of the message "End tag for li seen, but there were open elements" lets you know that something is wrong. It's up to you to figure out what it is. A good place to start is to check your container tags and make sure they are in pairs. In this case, that is the problem. You can scroll down to view the other errors. However, since multiple error messages are often displayed after a single error occurs, it's a good idea to fix one item at a time and then revalidate.
5. Edit the design.html file in a text editor and add the missing `</strong>` tag. Save the file. Launch a browser and visit **http://validator.w3.org** and select the "Validate by File Upload" tab. Click the Browse button and select your file. Click the Check button.
6. Your display should be similar to that shown in **Figure 11**. Notice the message, "Document checking completed. No errors or warnings to show." This means that your page passed the validation test. Congratulations, your web page is valid!

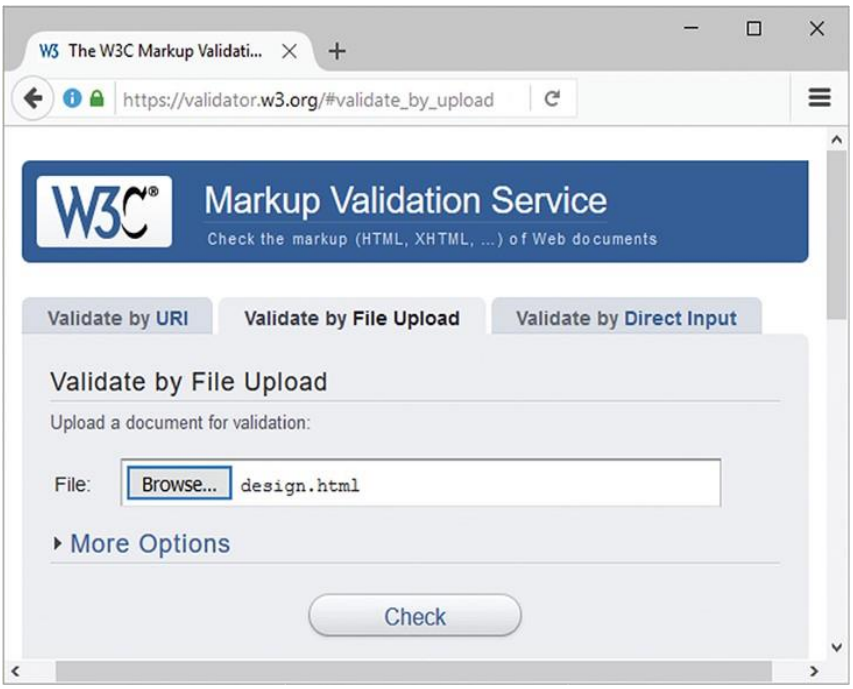

Figure 9 Validate your page. Screenshots of W3C. Courtesy of W3C (World Wide Web Consortium)

Figure 10 The error indicates line 12. Screenshots of W3C.

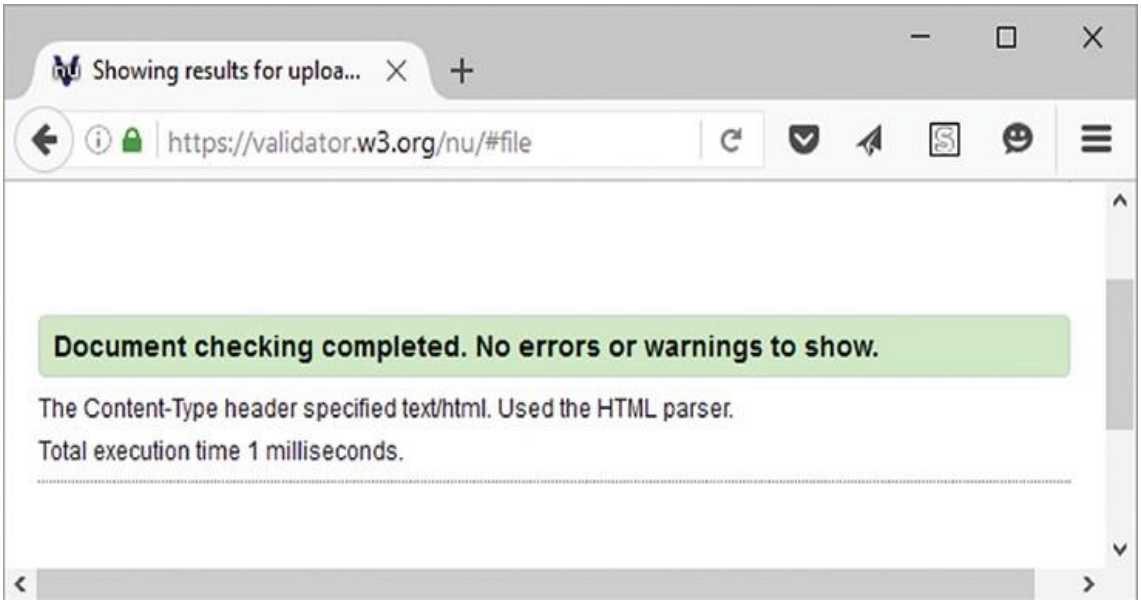

Figure 11 The page has passed the validation test.

Lab Completion / Submission:

Complete all the lab practices. Take the screenshots of your completed webpages; put them into a single word file and submit it to **Blackboard -> CP160 -> Assessments -> Lab 3**; due date: today

After-lab Assignment:

1. Create a one-paragraph conclusion of what you have learned during the lab today.
2. Markup language code alone does not make a web page—design is very important. Access the Internet/www and find two web pages—one that is appealing to you and one that is unappealing to you, and answer the following questions:
   A. What is the URL of the website?
   B. Is the page appealing or unappealing? List three reasons for your answer.
   C. Would you encourage others to visit this site? Why or why not?

Put 1 and 2 into a word doc, submit to **Blackboard -> CP160 -> Assessments -> Lab 3;** due date: 1 week from today.

Starting your project work:

You will complete a large course project as specified in the course outline. The course project requires you to create a multi-page responsive website using the skills and knowledge (HTML, CSS, and maybe JavaScript if possible) learned from this course. Your project work will be mainly evaluated from the following aspects, e.g., completeness, richness/number of features, novelty, clarity, contribution, amount of skills/techniques used, friendliness/accessibility, etc.

From this week, you should start thinking of your project proposal. As a student project, your website can be any. Some examples are, a simple ecommerce website, a portfolio website, a family website, an event website, an entertainment website, an educational website, a media website, a personal website, etc.

The first part of the project work is to make out a project proposal and a plan to complete each step. A good proposal and plan is critical to a quality project. You have long enough time (about 4 week) to work on the proposal/plan. The due date will be on Oct.21.

In this week, you can google, "web development projects", "website examples", "how to write website development proposals", or other similar words to get some fundamental ideas. More guidelines will continue to be provided in the following weeks for you to work on the project. As we continue to learn more skills, you will keep on improving your proposal and website.