JAVASCRIPT

152-097

Unit 12 Assignment

Introduction

In this lab you will create a JavaScript module to be used in a web application.

MAKE SURE download the **Unit12_Modules.zip** file from Blackboard.

Once completed, attach this completed word document to this assignment for grading.

Use the **Discussion Forum** if you have any questions regarding the how to approach this assignment. You can also email your instructor directly for assistance if you have any questions.

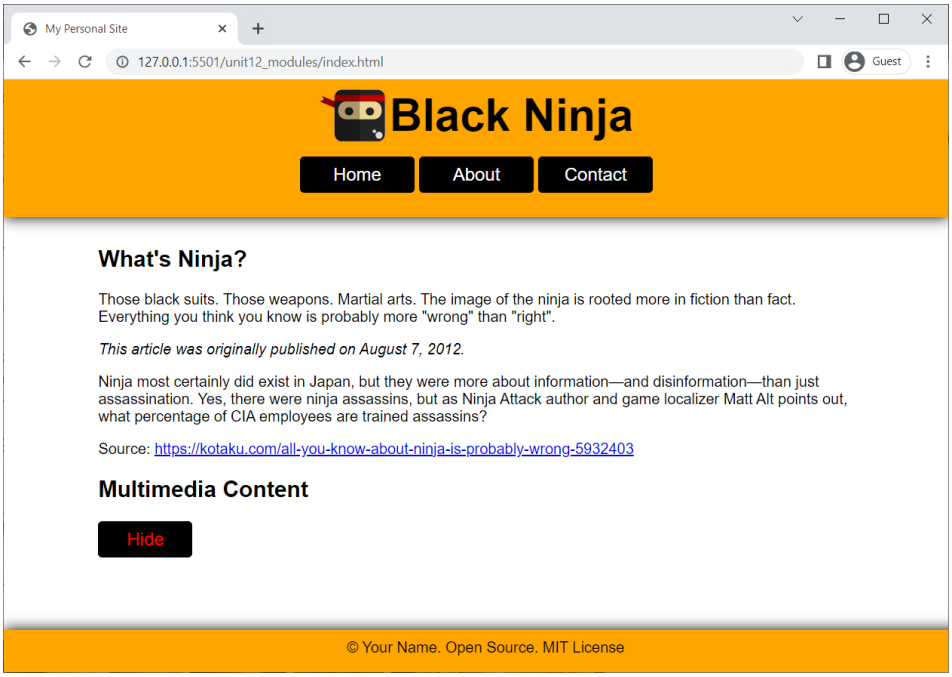
Submit your complete assignment in the unit *Apply* section of the course.

Instructions

You are to complete the following actions.

Update Author Name

1. Download the **Unit12_module.zip** file and extract it to your local drive.
2. Open the **unit12_modules** folder in your favorite IDE.
3. Run the index.html on the browser and navigate around to ensure it's working.



4. Examine every html file and notice that some elements are left blank:
    a. <h1 id="title"></h1> *(index.html only)*
    b. <nav></nav>
    c. <footer></footer>
    d. *(index.html only)*

    We'll use JavaScript to update them dynamically using DOM manipulation.

5. Open the **main.js** file and update the **author** with your name. This variable is used to be rendered in the footer of every page. This script also updates the navigation and footer immediately when it's loaded.
6. Save the file and verify that it's reflecting on the site.

Create Modules

In this section, we're going to create two JS modules called **media.js** and **home.js** to display multimedia content on the landing page (index.html). The media.js contains the content to be imported into the home.js module which then exposes it to the web page.

Part A: Create the media.js module

1. Open the index.html file and add the following tag to the head section of the page.

```
<script src="scripts/home.js" type="module"></script>
```

2. Save the file.
3. Open the media.js module. First, we will create an object of two properties (internal objects). One for the video and the other for audio. Study the code below.

```
const media = {
  video : { id:'video-lighthouse', src:'media/lighthouse.m4a' },
  audio : { id:'audio-podcast', src:'media/podcast.mp3' }
}
```

4. Next, we will create three methods (functions) to create the media, destroy the media, and toggle a show and hide feature.
5. Create a method called **createMedia**. Below code is using ES6 arrow function and the templating syntax to interpolate text. Make sure you're using the back tick symbols and not the single quote or the text interpolation won't work.

```
const createMedia = ( el ) => {
  const videoTag = `<video id="${media.video.id}" src="${media.video.src}" controls></video>`;
  const audioTag = `<audio id="${media.audio.id}" src="${media.audio.src}" controls></audio>`;
  el.innerHTML = videoTag + audioTag;
}
```

6. Create a method called **destroyMedia** to destroy/empty the media element.

```
const destroyMedia = ( el ) => {
  el.innerHTML = "";
}
```

7. Create a method called **showHide** to show and hide the media content. When the user clicks the button, it checks the status if the media's CSS "display" property. If it's set to "none" then

set it to “Block”, create the media content, and update the DOM. Otherwise, set it to “None”, destroy the media content, and update the DOM accordingly.

```
const showHide = ( el, button ) => {
  let status = el.style.display;
  if(status == "none"){
    el.style.display = "Block";
    button.innerText = "Hide";
    button.className = "red";
    createMedia( el );
  }else{
    el.style.display = "None";
    button.innerText = "Show";
    button.className = "white";
    destroyMedia( el );
  }
}
```

8. Finally, we export the **createMedia** and **showHide** methods. Everything else is private.

```
export {createMedia, showHide}
```

9. Save the file.

Part B: Create the home.js module

1. Open the home.js module.
2. First, let’s import the two functions from the media.js module so we can use it in this module.

```
import { createMedia, showHide } from "./media.js";
```

3. Next, create some DOM element references so we can update them.

```
const btnToggle = document.getElementById("btn-show-hide");
const media = document.getElementById("media");
const title = document.getElementById("title");
```

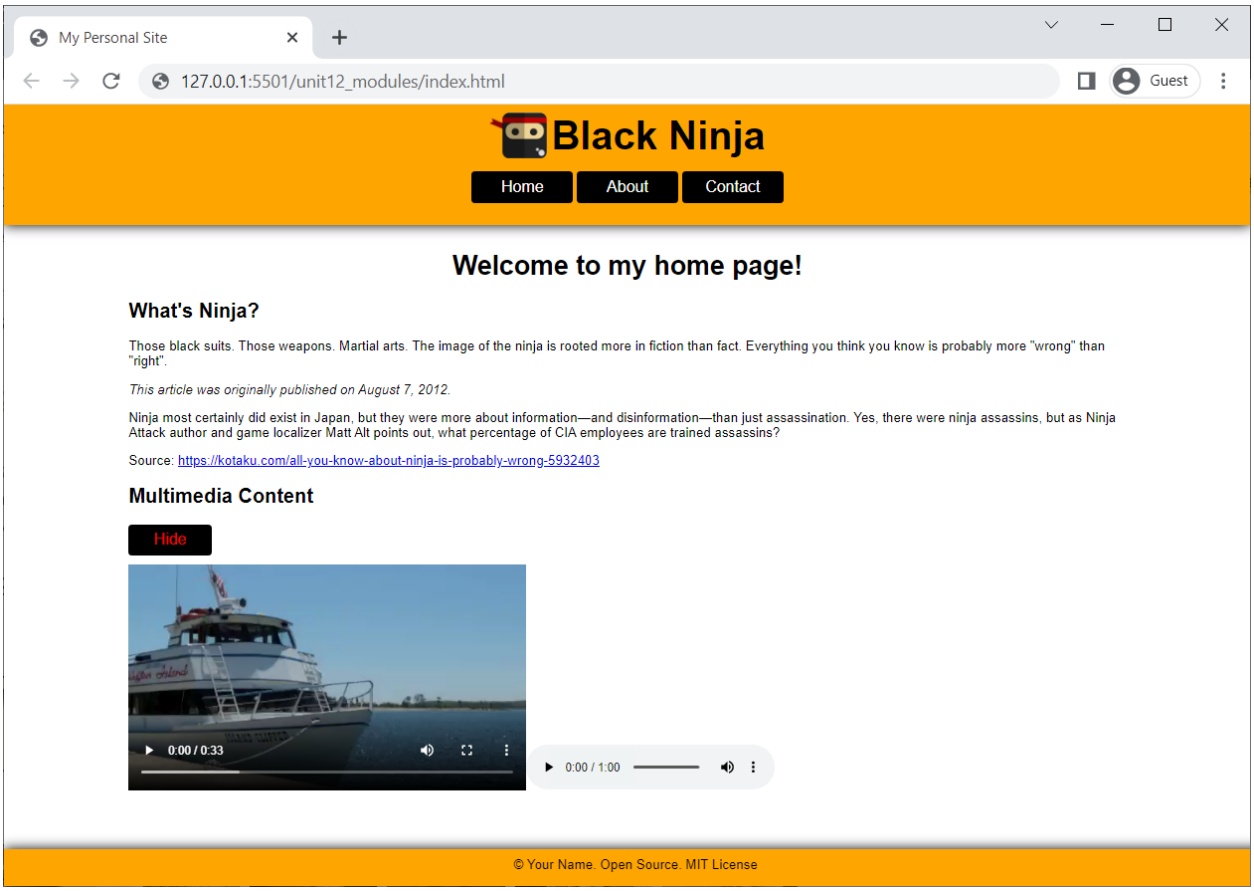
4. Update the title and media content immediately.

```
title.innerHTML = "Welcome to my home page!";
createMedia( media );
```

5. Finally, we'll register a click event to the button. When the user clicks on the button, it will invoke the **showHide** method we imported from the media.js module.

```
btnToggle.addEventListener("click", () => showHide( media, btnToggle ));
```

6. Notice how the **showHide** method is wrapped inside another callback function. This is how you invoke a function that takes parameters in an event. If you just call the **showHide(media,btnToggle)** method without wrapping inside the callback function, it will fire immediately and will error out. We want to call it in a future time (only when the user clicks the button).
7. Save the file.
8. Run the application and ensure that everything works.
9. Fix any errors.



Congratulations! You've completed the assignment.