

Homework

Homework



- For this homework, you will continue from the previous session's homework.
- Before you begin working, see the finish example [End Day 4 Homework](#) on today's session in GAP (Week 4 Day 1).
- Launch VS Code. Make sure [auto save](#) is on.
- Navigate to locate the [homework-module1](#) folder via *Open folder*.

Reminder:

Make at least 2 commits and push to your remote Github repository during and also at end of coding.

Homework



- Open [contact.html](#) in VS Code.
- First, we'll write the codes to create the two elements contain within this feature. Enter the following html codes within the form:

```
<div class="commentbox">
  <label>How did you hear about us:</label>
  <textarea></textarea>
</div>
```

```
<!--Random Codes-->
<div class="randomcode">
  <input type="text" name="randomcode" placeholder="Enter characters">
  <p id="codes"></p>
</div>
```

```
<div class="button">
  <input type="submit" id="submit" value="Inquire">
</div>
```

Homework

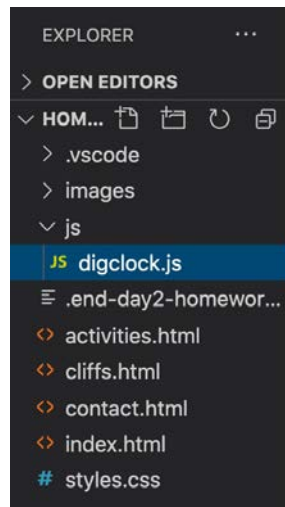


- Before you start writing more JavaScripts today, you are going to transfer all existing codes to an external document and then link it back to the HTML page – much like you did with the CSS codes in Week 2.
- Let's start with the index.html page. On the bottom of this page, highlight all the *digital clock scripts* within the script element (including the comments but excluding the <script> element). Cut the scripts out leaving the *partners script* and <script> element behind.
- On the explorer, create a new file. Name this file **digclock.js**. Paste the scripts on to this file.

Homework



1. On the explorer, create a new folder and named it **js** and move the **digclock.js** in to this folder.



2. Return to *index.html*. Add a new `<script>` element and above the partners scripts, enter the following script to link the js file to the HTML page:

```
        </div>
    </footer>
</div>

<!-- Digital Clock -->
<script src="js/digclock.js"></script>

<script>
    /* ----- PARTNERS ----- */
```

3. Preview the page to make sure things are still working the same.

Homework



- Next, you will transfer the partners scripts on all the pages to one external js file.
- Go ahead create a new file in explorer in VS Code and name it as [site_scripts.js](#). Save this file in the [js](#) folder.
- On the *index.html* page, highlight the *partners scripts* including its comments but excluding the `<script>` element. Cut the scripts and paste on to [site_scripts.js](#).
- On *index.html*, link [site_scripts.js](#) file to the page:

```
<!-- Digital Clock -->
<script src="js/digclock.js"></script>

<!-- Partners -->
<script src="js/site_scripts.js"></script>

</body>
```

Homework



- On *activities.html*, *cliffs.html* and *contact.html*, remove the same partners scripts including the `<script>` element. You won't need this same codes as its already saved in the [site_scripts.js](#) file.
- All you to do next is link the js file on these HTML pages, like what you did in the previous step. Copy the script from the *index.html* page and paste it on to the other three html pages.

```
<!-- Partners -->  
<script src="js/site_scripts.js"></script>
```

- Preview all pages to make sure things are still working the same.
- You are now ready to start today's JavaScript homework. But first, create a new file, name it as [randomcodes.js](#) and save it in the [js](#) folder.

Homework



- In this homework, you will add a code verification feature on the contact form on the contact page. This verification feature will consist of two elements:
 1. A text input field for user to enter eight characters.
 2. A string of eight characters.
- Below is an example of what you will be creating:

@8UEci7p

Homework



- Next, you will write the JavaScript needed to get this to work. We'll start by creating a function to wrap our scripts. Why a function? Because this allow us the flexibility to call the scripts when needed or in this case allow us to insert the result or outcome from the scripts to display on the body of the page. Enter the following script on the [randomcodes.js](#) file to create the function:

```
/* ----- RANDOM CODES ----- */  
  
//Function to generate combination of characters  
function generateCode() {  
  
}
```

Homework



- Within this function, we will start with a couple of variables to store some information now and also down the road later. One will be used to store the generated codes or characters. The other will be used to store the type of characters we want our generated codes or characters to show. Enter the following scripts within the function:

```
//Function to generate combination of characters
function generateCode() {
    //Create variables to store generated codes and the type of characters we want to show as codes
    var code = ' '; //initialize to null value
    var str = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789@#$',
}
```

Homework



- Next, we want to randomly generate 8 characters but one at a time. To help us do that, we'll use a loop, but more specifically, the **for** loop. The **for** loop is ideal for this because it's design to count through a set of steps or times. Enter the following scripts below the previous variables:

```
//Function to generate combination of characters
function generateCode() {
    //Create variables to store generated codes and the type of characters we want to show as codes
    var code = ' '; //initialize to null value
    var str = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789@#$',

    //Generate character multiple times using a loop
    for (i = 1; i <= 8; i++) {
    }
}
```

Homework



- Within the loop, we will write two lines of scripts. The first script is to randomly generate a character from the variable that contains the type of characters and store in a new variable. The second script will use that character generated and accumulates in to the empty variable we created at the beginning and this process will repeat eight times. Enter the following scripts within the loop:

```
//Generate character multiple times using a loop
for (i = 1; i <= 8; i++) {
    var char = Math.random()* str.length; //random select a character from the variable and then store in a new variable
    code += str.charAt(char) //accumulate the generated character into a string of 8 characters
}
```

- And to finish off this function, we will add a script to return the final result that contains the accumulated characters:

```
//Generate character multiple times using a loop
for (i = 1; i <= 8; i++) {
    var char = Math.random()* str.length; //random select a character from the variable
    code += str.charAt(char) //accumulate the generated character into a string of 8 characters
}
return code; //return the final accumulated string when loop ends
```

Homework



- Now that the scripts to generate a random character and then put together as a string of characters, it's time to display this string on to the page.
- Enter the following code below and outside of the function:

```
    return code; //return the final accumulated string when loop ends  
}
```

```
//Get HTML element to display  
document.getElementById("codes").innerHTML = generateCode();
```

Homework



- This next step has nothing to do with the characters we generated, but more of a feature that we want to use a bit later. This next script will disable the submit button to make it inactive. We will revert it back to active in the next lesson.
- Enter the following code below the last script:

```
//Get HTML element to display
document.getElementById("codes").innerHTML = generateCode();

//Disable Button
function disableButton() {
    document.getElementById("submit").disabled = true;
}
```

Homework



- Next is to activate the function that disable the button. Enter the following script below the function:

```
//Disable Button  
function disableButton() {  
    document.getElementById("submit").disabled = true;  
}
```

```
//Activate function  
disableButton();
```

Homework



- The button will not visually show it's disabled unless we switch off any CSS style on the button. In this case, we will switch off the background color.
- In the *styles.css*, scroll to the bottom of the file and deactivate the background color as follows:

```
form .button input {  
  width: 8em;  
  height: 2.5em;  
  font-size: 1.2em;  
  font-weight: bold;  
  /*background-color: #4977D1;*/  
  color: ■ #ffffff;  
  border: none;  
}
```

When preview on the browser:

How did you hear about us:

Homework



- **Due Date:**
 - This Sunday 10.30PM PT.
- **Submission**
 - Post Github URL link on GAP Week 4 Day 1 Homework dropbox

Questions?



Connect with Us (WEB301/501/801)

Professor

Rich Loke

richloke@westcliff.edu

Teaching
Assistant

Christopher Paul

christopherpaul@westcliff.edu

Teaching
Assistant

Ilya Valasov

ilyavalasov@westcliff.edu

Teaching
Assistant

Elizabeth Kipp

elizabethkipp@westcliff.edu

End of Presentation