Homework

Homework

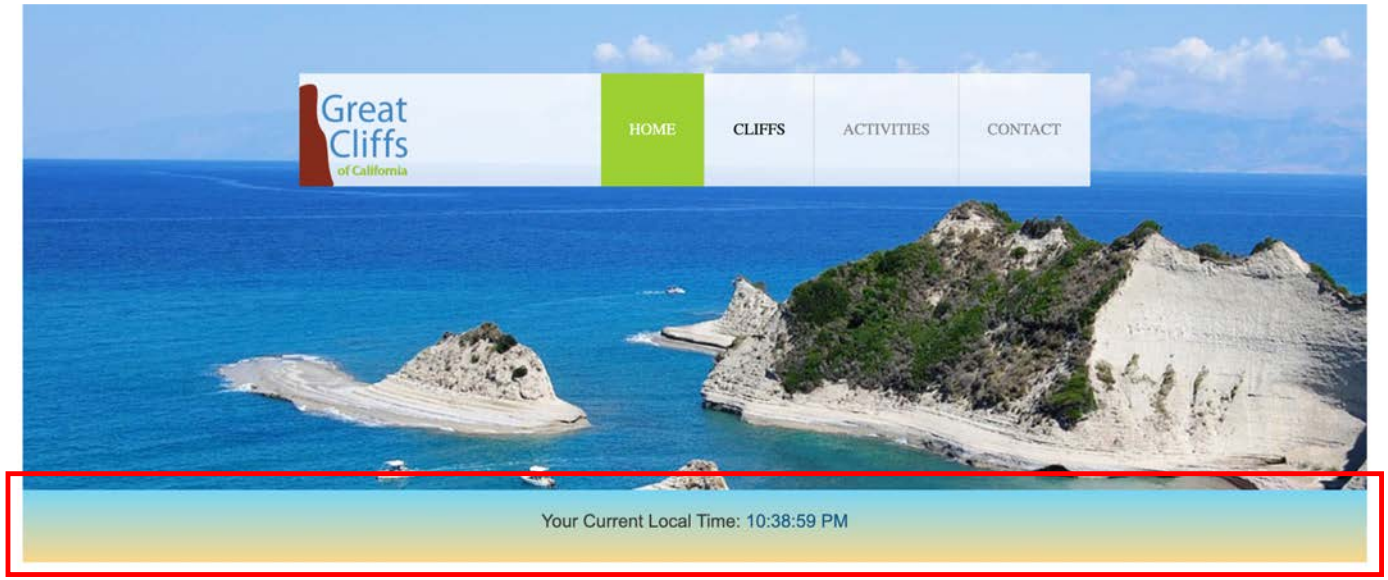
- For this homework, you will continue from the previous session's homework.
- Before you begin working, download the End Day 1 Exercise from today's session in the lesson portal on GAP and view the finish example.
- Launch VS Code. Make sure auto save is on.
- Browse to locate the **homework-module1** folder via *Open folder.*

Homework

- In this JavaScript homework, you will add a digital clock on the home page. This digital clock will run continuously and it's based on the current local time of the user.
- To create this digital clock, here is a quick rundown of the items we need:
  1. A HTML element specifically to display this clock along with a gradient background.
  2. A HTML element to wrap the JavaScript codes.
  3. JavaScript Concepts:
     a) Functions
     b) Variables
     c) Conditional statements
     d) HTML DOM
     e) Timer Events

Homework

- Below is what we want to achieve upon completing of this homework:



Homework

- Let's begin by adding a HTML element to display the clock and background.
- Open index.html. Add the following codes:

```
</header>
<section id="timeofday">
    <span>Your Current Local Time: </span>
    <span id="clock"></span>
</section>
<main>
```

- Next, scroll to the bottom of the page, and add this HTML element to wrap the JavaScript codes:

```
        </footer>
</div>

<!--JavaScript Codes-->
<script>

</script>

</body>
</html>
```

Note: Why the bottom of the page? Because JS are not needed until the page is fully loaded.

Homework

- The very first JavaScript we want to write is to create a function to wrap the codes for the digital clock. Why do we need a function? Function can help 1) not run codes until you need it. 2) repeat/rerun codes as many times as you wish.
- In the digital clock case, for the latter reason. Enter the following within the <script> element:

```
<!--JavaScript Codes-->
<script>
    //Create a time data function
    function currentTime() {

    }
</script>
```

Homework

- Next, we'll create a bunch of variables within the function. Each variable will store its respective data/value. The first variable is to get the **current date** from the user's computer. This date contains multi-piece information ie. year, month, day and time. The next set of variables will store data extracted from the date stored: **hour**, **minutes** and **seconds**. And finally a variable to store custom string data: **AM** or **PM**. Enter the following within the function:

```
//Create a time data function
function currentTime() {
    //Declare variables
    var d = new Date(); //Get current date
    var hr = d.getHours(); //Get current hours
    var min = d.getMinutes(); //Get current minutes
    var sec = d.getSeconds(); //Get current seconds
    var ampm; //Declare empty variable to store AM or PM
}
```

Homework

- The time data by default do not display single digits of hour, minutes and seconds with a zero in front of the digit. Example: for minutes and seconds, we want it to display: 00 or 09. But for hour, 9 instead of 09 is what we are looking for.
- Therefore for the next scripts, we will need conditional statements to check if the minutes and seconds are single digits and if they are, we will add a 0 in front of it. Enter the following below the last variable:

```
var sec = d.getSeconds(); //Get current seconds
var ampm; //Declare empty variable to store AM or PM

//Add 0 to single digits for seconds
if (sec < 10) {
    sec = "0" + sec;
}
//Add 0 to single digits for minutes
if (min < 10) {
    min = "0" + min;
}
```

Homework

- Next, we want to display AM or PM after the digits. To do this we have to get the current time. For example, if time is before 12:00PM or noon (from 0 to 11) it will AM and anything past 12:00PM (12 to 23) it will be PM.
- The only problem is the default time format is in military time ie. 0:00 (12:00AM) or 23:00 (11:00PM). To display in regular time format, we will write scripts to do the conversion.
- The scripts will check if the current hour is past the 12:00PM mark. If it is, we will deduct it from 12. Example: if the current time is 15:00 (which is greater than 12:00PM mark), we will deduct 12 from it which will give us 3 as in 3PM.
- So therefore, there are two parts to this script: 1) check whether to assign AM or PM and 2) convert military hour to regular hour if it exceeds 12.

Homework

- Enter the following scripts below the last if statement:

```
//Add 0 to single digits for minutes
if (min < 10) {
    min = "0" + min;
}
```

```
//Determine AM or PM string
if (hr == 12) {
    ampm = "PM"; //Set to PM
} else if ( hr > 12) {
    hr -= 12; //Deduct 12 from hours greater than 12 (military time)
    ampm = "PM"; //Set to PM
} else {
    ampm = "AM"; //Set to AM
}
```

Homework

- Now that we have all the necessary time and string data (stored in variables) to display our clock, we can now write a script to assemble them together to form the time display format we are going after.
- First, we will store what we assembled in a variable so that it can simply be called when we want the clock to run. Secondly, our assembly will consist of variables and strings (“ ”) put together with the help of the concatenation symbol +.
- Enter the following script below the last if statement:

```
} else {
    ampm = "AM"; //Set to AM
}

//Assemble time format to display
var time = hr + ":" + min + ":" + sec + " " + ampm;
```

Homework

- Next, you will write a script to display the assembled clock format on the page.
- If you recalled, earlier you'd created a HTML element to display this clock. We will write a HTML DOM script to tell it to display within that new element. And, don't worry for now what this specific code in detail means, as you will learn this next week. For now, you will use it to display the clock, enter the following below the assembly code:

```
//Assemble time format to display
var time = hr + ":" + min + ":" + sec + " " + ampm;

//Display current local time and time zone on HTML elements
document.getElementById("clock").innerText = time; //adding time
```

Homework

- The foundation codes for the clock is complete. They are wrapped in the function currentTime. However, as you know functions don't auto run unless you invoke or call it, and you call it by the name of the function, which in this case is currentTime. The calling is typically done from outside of the function, preferably after the function codes.
- Enter the following below where the function ends:

```
        //Display current local time and time zone on HTML elements
        document.getElementById("clock").innerText = time; //adding time
    }

    //Initial run of time data function
    currentTime();
</script>
```

Homework

- Preview on the browser. You will notice the clock did display but other than that, it is not ticking forward. Here's why: all the scripts within the function though is called to run, is only running once. That's how it is. But of course there's a way to fix this. We can resort to a JavaScript concept called **Timer Events**.
- The general concept of Timer Event is that it repeats its occurrence every so often, usually in seconds. This method takes two parameters: a *function* and wait *time in milliseconds* to repeat. Example:

  timerEvent(*function*, *milliseconds*)

- There are two key Timer Event methods:
  - setTimeout()
  - setInterval()

Both methods are very similar in their execution. So any one of them can be used in most situations.

Homework

- For this homework, you will use the **setInterval()** method. The parameters to be taken by the method are:
  - currentTime function
  - default 1000 milliseconds (1 second; 1000 milliseconds = 1 second)
- Enter the following within the function, at the very bottom after the display scripts:

```
//Display current local time and time zone on HTML elements
document.getElementById("clock").innerText = time; //adding time

//Run time data function every 1 second
setInterval(currentTime, 1000); //setting timer
```

- Preview page on the browser before proceeding to the final steps. The clock should now tick continuously.

Homework

- The final of the homework is to style this display area to make it more presentable.
- Currently:

- After styling:

Homework

- This part of the homework will be on you. Using your CSS knowledge, skills and research, style this display area as seen on the previous slide.
- Using CSS comments, create a home area within your stylesheet and write your codes there.
- Below are styles you will need:
  - **Font:** Arial, Helvetica, sans-serif
  - **Font colors:** #3a3a3a, #125688
  - **Linear gradient color:** 0deg, rgba(251,215,139,1) 0%, rgba(118,213,245,1) 100%
    - Get gradient color from here: https://cssgradient.io

Homework Challenge

- Add a time zone to the clock as shown below:

- You will need additional variables:

```
var utchr = d.getUTCHours(); //Get current Greenwich Mean Time (GMT)
var timeDiff; //To store time difference between GMT hour and Local hour
var adjTimeDiff; //To store time difference converted to positive number
var timeZone; //To store the 4 time zones (PT,MT,CT,ET)
```

Homework Challenge

- Here are the things you need to do (in that order):
  - Convert Greenwich Mean Time from military time to standard time
  - Calculate time difference between GMT hour and local hour
  - Convert time difference, if negative, to positive (adjusted time difference)
  - Check which time zone based on the converted time difference between GMT Hour and Local Hour
  - Add time zone to the very end of the assembly time format

Homework & Homework Challenge

- **Due Date:**
  - Due before the next class session starts.
- **Submission**
  - Submit **homework-module1.zip** in the homework dropbox for Week 3 Day 1 on GAP.

Questions?

Connect with Us (WEB801)

Rich Loke
richloke@westcliff.edu

Reshae Alagbada
reshaealagbada@westcliff.edu

Elizabeth Kipp
elizabethkipp@westcliff.edu

Jonas Restad
jonasrestad@westcliff.edu

Slack: spring2021s3-bootcamp-mon

Connect with Us (WEB301 & WEB501)

Ghani Zahid

ghanizahid@westcliff.edu

Ilya Valasov

ilyavalasov@westcliff.edu

Christopher Paul

christopherpaul@westcliff.edu

Nico Lauria

nicolauria@westcliff.edu

Yi Sun

yisun@westcliff.edu

Slack: spring2021s3-bootcamp-tue

End of Presentation