**PLANETS**

**Objectives:** JavaScript, DOM

**Task:** Create a webpage with JavaScript functionality to click through a list of the planets in our solar system. There will be buttons to go to the next and previous planet.

**Build Specifications:**

Set up a webpage with three files: index.html, styles.css, script.js. Link the CSS and JavaScript to the webpage. Remember: CSS is linked in the <head>. JavaScript is linked at the end of the <body>.

*Phase 1: Create the planets array.*

* This should be an array of eight objects. Each object has name (string), inner (boolean), diameter (number), and color (string) properties. Use the following values.

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **inner** | **diameter** | **color** |
| Mercury | true | 3031.9 | #696969 |
| Venus | true | 7520.8 | #b89165 |
| Earth | true | 7917.5 | #5a5b86 |
| Mars | true | 4212.3 | #df8c4e |
| Jupiter | false | 86881 | #f6a049 |
| Saturn | false | 72367 | #dcd3a1 |
| Uranus | false | 31518 | #b4d9df |
| Neptune | false | 30599 | #456eff |

* Write a loop to log the name of each planet in the array.
* Write a function that takes in the array of planets as a parameter and returns the average diameter of planets in the array. Call this function and log the result to the console.
* Here's an example of the output you should see in the console.  
  Graphical user interface, application

  Description automatically generated

*Phase 2: Previous and next buttons*

* Add to the HTML two <button>s and a <p> which will hold an index number, like this:  
  Graphical user interface, text, application, chat or text message

  Description automatically generated
* In JavaScript, create a variable named **index** which is set to **0**.
* Add event handlers to the buttons so that the index variable is incremented or decremented and the <p> is updated with the new number. The next button adds one to index. The previous button subtracts one from index.

Phase 3: Display planet information

* Add an <h2> for the planet name, a <p> for "Inner planet" or "Outer planet", and a <div> to represent a visual circle of the planet.
* Add CSS to make the planet appear round and to add a nice transition. Use these properties.  
    border-radius: 50%;

  transition: all 200ms;

* Create an updatePlanetDisplay() function that uses the index select planet information from the array and update the text and styles of the HTML. Call this function from the button event handlers after the index is changed.
  + Update the text of the <h2> using the planet's name.
  + Update the text of the <p> to "Inner planet" if the planet's inner is true or "Outer planet" if it is false.
  + Update the background-color style of the <div> to the planet's color.
  + Update both the width and height of the <div> to be the planet's diameter divided by 100. Add "px" to the end.
* Also in updatePlanetDisplay()...
  + Disable the Previous button when the index is 0.
  + Disable the Next button when the index reaches the maximum index (array length - 1)
  + Make sure the buttons are enabled otherwise.
* Also call updatePlanetDisplay() as soon as the page loads to load up the initial information for the first planet, Mercury.

**Sample Screenshots:**

Graphical user interface, text, application, chat or text message

Description automatically generatedA picture containing graphical user interface

Description automatically generatedShape

Description automatically generated