JS 6

A Calendar App must have a input area for the user to type in their event data. This input area will look like a search bar. When the user is finished typing in their event data, there must be a button that the user "clicks" on to signal an event. This event will then activate a JS function to check the data that the user entered. The check is to make sure that the user entered valid data.

We continue with events and action by connecting HTML with JS using "onclick". **This is an important session because we are using a mouse click to transfer data from HTML to JS**.

1. using input using search bar 2. valid data detection 3. if statement 4. event and action

NOTE: If we open something, we MUST close it. The slash (/) is used to close a tag.

```
<p  id = ”p2”  > Hi       </p>
<h2 id = ”subT”> Sub Item </h2>
```

==

Continue to the next page for instructions

1. Write JS code in between <script> </script>. ONLY write the code in RED and Purple.

```
var p2DOM = document.getElementById("p2");

p2DOM.style.color = "#ccdead";
document.getElementById("p2").style.fontFamily = "Arial";
document.getElementById("p2").style.fontSize = "larger";


function iDV()
{
      var uInput = document.getElementById( "nEInput" );

      var eData = uInput.value;

      alert( "user entered: " + eData );


      var ePTag = document.getElementById( "ePTag" );
      ePTag.innerHTML = eData;

}
```

Explanation of JS Code

We are writing a function definition called iDV(). This is an action and we will link this action to HTML later on. Inside the function definition, we get input from the search bar and display in as an alert and dynamic data.

Remember that if we want to update the website with dynamic data, we use
`document.getElementById( "ePTag" );`

This allows JS code to **link up** with HTML. In this case, the JS code is looking for the HTML with id "ePTag". Next, we use `.innerHTML` to change the content by using the assignment operator, which is the single equal sign.

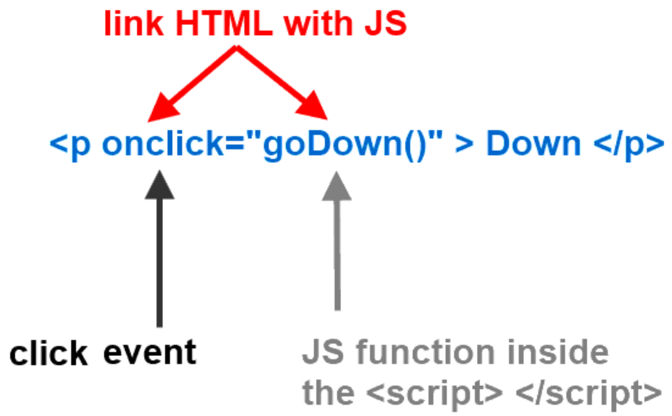
2. Write HTML code in between <body> </body>. ONLY write the code in SKY BLUE and Purple.

```
<p id=”p2”>Input From User</p>

<button
id      = ”nEventB”
onclick = ”iDV()”
>
Normal Event
</button>


<input
id        = ”nEInput”
value     = ”first input”
minlength = ”20”
maxlength = ”50”
name      = ”nEInput”
/>

<p id = ”ePTag”>
The paragraph above was changed by a script.
</p>
```

Explanation of HTML Code

There is a new HTML tag that is very powerful. The new HTML tag is the `<input />` tag. This is basically a search bar that allows user to enter **ANY** type of data **(MAGIC later on!!!).** Look at it and see that it has an id of "**nEInput**". Since we gave it an id, JS is able to find it **(MAGIC later on!!!).**

Next, look at the "**value**". This is the data that the user entered in that search bar. We will use JS to take this data and use it for color change (**MAGIC later on!!!**)



```
<p onclick="goDown()" > Down </p>
```

**When we "click" on the word "Down",
the JS function goDown() is called**

Remember that we can generate an event by using "on". In this case, the event is **onclick**. When we click on the button "Normal Event", we activate the event and the action **iDV()** is called.

Remember that the action **iDV()** is a Javascript function that we ALREADY wrote up there in between the **<script> </script>**

Click the green "Run" button.

Type something in the search bar and click on the button "Normal Event". You will see that any data that we enter in the search bar will be put as an alert **AND** also in the website. This is how Gmail and all other websites are able to check the username and password.

Text Message and Chat - Echo

Have you ever wondered how text messages, chat boxes, and comment boxes work? No need to wonder anymore. You guys just did it!!! A text message and a chat message is basically an echo function.

Echo means to take the user input and display it back. You guys just did that by taking in input and displaying it back to the website. You guys just created the echo function of text message, chat boxes, and comment box.

3. Write CSS code in between <style> </style>. ONLY write the code in GREEN

```
p#p2
{
    font-weight   : bold;
    border-bottom : 3px solid #606060;

}

input#nEInput
{


}


button#nEventB
{
    color    : #006400;
    border   : 2px solid #f69a89;

}
```

Explanation of CSS Code

We are styling the button and the title.

CSS Challenge. Write the code in between the <style> </style>. ONLY write the code in GREEN

1. when we :hover over the button#nEventB, style it as

```
color    : #006400;
border   : 2px solid #606060;
background-color: #e0e0e0;
```

2. style the body

```
border   : 5px dotted #800000;
background-color: #DAA520;
```

3. notice that the style of the search bar is empty. Inside `input#nEInput,` style it any way you like. Change the font-size, font-color, and etc.

4. Write JS code in between <script> </script>. Time for MAGIC!!! Write the code in RED. Type ffd700 into the search bar and press "Normal Event"

```
function iDV()
{
      var uInput = document.getElementById( ”nEInput” );

      var eData = uInput.value;

      alert( ”user entered: ” + eData );

      /* -- color change via input text box

      */
      var cValue = document.getElementById( ”nEInput” ).value;
      alert( ”cValue: ” + cValue );
      document.getElementById(”mBody”).style.backgroundColor = ”#” + cValue;

}
```

Explanation of CSS Code

JS uses id to look for HTML code and the id allows JS to link up with HTML. JS is able to find the search bar because we gave it an id of "**nEInput**". The `.value` is a variable that holds the data that the user entered in the search bar. We use this data to change the background color of the `<body> </body>` tag.

Notice that we gave the body an id of "**mBody**" and this is how JS is able to find the <body> and change it's background color.

JS Challenge. Write JS code in between <script> </script>.

Notice that if we put the search bar as empty and click on "**Normal Event**", it will also accept **BLANK DATA**. It is probably not a good idea for a Calendar App to accept blank data. There is a way to fix this and the solution is to use the conditional statement.

Remember that one way to implement a conditional statement is to use the "if" statement. For this challenge, check if the eData is equal to "".

If TRUE, alert the user that "Blank data detected, try again".

ELSE, accept the valid data and display it onto the website using .innerHTML

JS Challenge 2. Write the code in between the <script> </script>

What if someone types in the search bar the data "...". This data will cause the JS code to accept. Your task now is to write ANOTHER conditional statement to check if eData is equal to "...".

If TRUE, alert the user that "Try again".

ELSE, accept the valid data and display it onto the website using .innerHTML