

Lesson 11 – Intro to JavaScript – Functions

Functions in JavaScript serve the same purpose as any other language. You can create a function that executes a script in a div or semantic, and you can also create a function that will respond to an event like onclick, onload, on mouseover, on mouseout, and even onkeypress. These will execute when the mouse action is recognized. For example, if I want something to happen as the user clicks on a button, and the event to the html tag and create the function to correspond to the event in the js file. Or if something is to happen as the form loads, add the onload function in the js file. Have a look at the following videos.

<https://www.youtube.com/watch?v=FOD408a0EzU> – Basic Functions.

<https://www.youtube.com/watch?v=HpbHoOwuW2I> – Calling event functions.

Creating a single .js file for all functions on this page.

Create a single js file for a page that will contain all the functions for that page. If you are working on the index.html page, for example, create a js file called index.js. The file can be defined in the head section just below the css link. Now when you want a function, place it in the js file.

```
<link rel="stylesheet" href="styles.css" />
<script src = "index.js"></script>
```

We have used templates for starting html and css files, you can also include the template for the js file as well. The file will contain a few statements that are required for many operations. Now when you create a new project, start with all three files in the folder.

```
// Desc:
// Author:
// Dates:

var $ = function (id) {
  return document.getElementById(id);
};

// Define format options for printing.
const cur2Format = new Intl.NumberFormat("en-CA", {
  style: "currency",
  currency: "CAD",
  minimumFractionDigits: "2",
  maximumFractionDigits: "2",
});

const per2Format = new Intl.NumberFormat("en-CA", {
  style: "percent",
  minimumFractionDigits: "2",
  maximumFractionDigits: "2",
});

const com2Format = new Intl.NumberFormat("en-CA", {
  style: "decimal",
  minimumFractionDigits: "2",
  maximumFractionDigits: "2",
});

// Start function definitions here.
```

Note the first lines below the comments. This is an option that allows you to reference each html element by its id. So now when you create an html tag that is needed in the js file, you need to add an id property to the tag. For example,

```
<button id="btnView" onclick="ShowStuff()">Click Me!</button> - This creates a button with the text Click Me!,  
and will execute the ShowStuff() function when clicked.  
  
<p id="txtShowReceipt"></p> - This will create a landing place for the receipt to be displayed.
```

Now the final step is to write the function that corresponds to the function call in the js file. For example if I want to display the message “Hello World!” when I click the button, prepare the function as follows:

```
function ShowStuff() {  
    // Show the message as the user clicks on the button.  
  
    document.write("Hello World!");  
}
```

Notice a problem in that the message seems to appear on a new page – we want it to appear in the div below the button. To do this, add a `<p id="ShowMsg"></p>` tag with an id as shown below the button. Then we will reference the id of the paragraph, and the text will appear where we want.

In the html page below the button, add the `<p>` tag as follows:

```
<p id="ShowMsg"></p> - This will create a landing place for the message to be displayed.
```

Now we can add the following code in the function to have the text appear where we want.

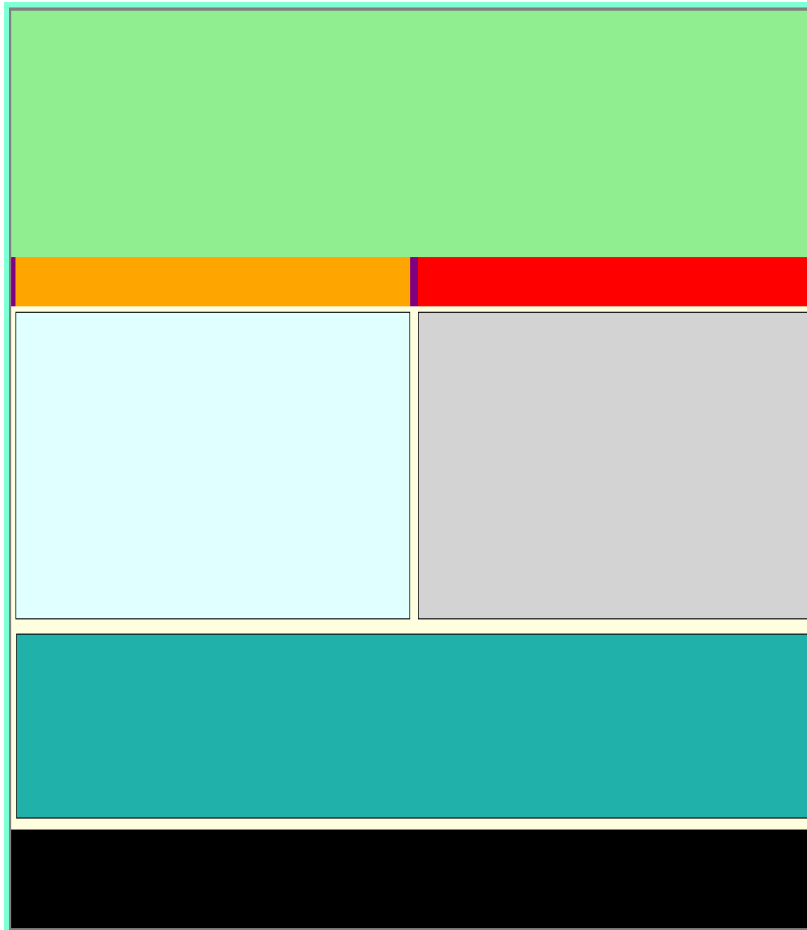
```
function ShowStuff() {  
    // Show the message as the user clicks on the button.  
  
    Msg = "Hello World";  
    document.getElementById("ShowMsg").innerHTML = Msg;
```

In this code, you create a variable that contains the message – you can concatenate literals and variables in the message variable as well.

Then we use the `getElementById()` and specify the id as assigned to the p tag, and assign the variable that contains the message. The message should now be displayed in the `<p>` tag.

Function Practice Exercises

Use the following interface (provided) to practice functions.



As you plan your code, identify if the function will just display the text as the page is loaded – in this case you will just create a function the determines and displays the text – or whether the content will display based on the click of a link or button – in which case you will write a function based on the click event of the button or link.

- Place a picture (Header.jpg with a width of 996px and a height of 200px in the header section.

- In the Nav section add links **Home** **About** **Contact** in the left div of the nav. Add a greeting to the right div and have it right aligned. The greeting will read “Good Morning / Good Afternoon / or Good Evening / or Good Nightbased on the current time. Get the hours associated

with the current date using getHours(). This is a 24-hour clock. Good Morning, is if the hour is between 6 and 12, Good Afternoon, is the time is between 12 and 18, and Good Evening, is the time is between 18 and 24, and Good night is the time between 0 and 6.

- In the mainleft div, add the links **View 1** **View 2** **View 3** **View 4** centered across the top of the div. As the user clicks on the links, display a quote for the user. Each link will display a different quote.
- In the mainleft div in the main section, add a centered button the reads “Button 1”. If the user clicks on the button, prepare a function for Computer Liquidations to gather data on new inventory items they bring into their store. Input the item name, the item cost (the amount the company paid for the item), and the number in stock. Calculate the retail price (the amount on the sales tag) by adding a markup of 75% to the item cost. Also determine the sales price of the item with 10% off, 25% off, 33% off, and 50% off – Note that this is the cost after the discount. Display all calculated values to the screen in a lovely table for the user.
- In the mainbottom div, either add a button or just call a function that displays something for the user. You can prompt for input, do some calculations, and display some results.
- Create a new page for the About option from the menu. As the page is displayed, include an image on the page in the main section. Write js code to create a slideshow based on that image. (Explained on the next page.

Create a Slide Show with a series of images

Create the image where you want it to appear. Most importantly, make sure that you add the id property in the tag. Load the first image into the img to start.

```
<img src = "Images/jan.jpg" id = "ImgSlide" width = "700" height = "500" />
```

Next move to the js file and add the following:

First step is to create an array and add the images to each element. Also create a variable that will be used to keep track of the index of the image array and initialize it to 0.

```
let step = 0;
let Images = new Array();
Images[0] = "Images/Jan.jpg";
Images[1] = "Images/Feb.jpg";
Images[2] = "Images/Mar.jpg";
Images[3] = "Images/Apr.jpg";
Images[4] = "Images/May.jpg";
Images[5] = "Images/Jun.jpg";
Images[6] = "Images/Jul.jpg";
Images[7] = "Images/Aug.jpg";
Images[8] = "Images/Sep.jpg";
Images[9] = "Images/Oct.jpg";
Images[10] = "Images/Nov.jpg";
Images[11] = "Images/Dec.jpg";
```

Next, create an onload event for the page. This event will run as the page is being loaded. It will call a function that will cycle the images.

```
window.onload = setInterval(gallery, 3000);
```

Finally, create the function called in the onload event.

```
function gallery() {
  //change image
  document.getElementById("ImgSlide").src = Images[step];
  //Or you can use - document.images.slide.src=Images[step];
  // is step more than the image array?
  if (step < Images.length - 1) {
    // No - add 1 for next image.
    step++;
  } else {
    // Yes - Start from the first image
    step=0;
  }
}
```