Exercise 16 JavaScript

Section 1. Your First JavaScript Commands

You are going to practice your JavaScript skills on a new "Seahorse" Aquamaniacs page. Let's get that page set up now and then write some JavaScript code!

To begin, find the Exercise16 folder on the zip file and copy Exercise16 folder to the Exercise folder on your computer. Copy the files from the folder into your "Aquamaniacs" folder, and confirm that the following files are placed in the correct locations:

- "Aquamaniacs/seahorse.html"
- "Aquamaniacs/PagePhotos/seahorse.jpg"
- "Aquamaniacs/cuttlefish.html"
- "Aquamaniacs/PagePhotos/cuttlefish.jpg"

Note: copy the files only, not the folder.

Load the "seahorse.html" file in your web browser to see the default content. You should see a captioned photo on the left and some paragraph text on the right.

Now, open "seahorse.html" in Komodo Edit go down to the <footer> element at the bottom. Add a new <script> element as shown below to output the current date and time to the footer.

```
<footer>
  <script>
    document.write("Today is " + Date() + "<br />");
  </script>
  <small>&copy;  Give it a name. All Rights Reserved.</small>
</footer>
```

Notice we added a line break
 at the end to separate the two lines.

Save your changes to "seahorse.html" and load the file in your web browser. Does it show the current date and time correctly?

You can practice writing other content and elements to the page with new <script> elements if you like. They can be placed anywhere in the <body>, including the "MainContent" area.

Section 2. JavaScript Events and Functions

You are going to practice your JavaScript skills on a new "Seahorse" Aquamaniacs page. Let's get that page

Let's add a button to our "Seahorse" page and display a JavaScript alert when the button is pressed.

Run Komodo Edit and load your "seahorse.html" page. Add a new button and JavaScript function at the bottom of the "MainContent" area as shown below.

```
<p>This odd creature is covered with bony plates ...

    <script>
      function wakeUp()
      {
        // display an alert for the user
        alert("OK, OK, I'm awake.");
      }
    </script>
    <div>
      <button type="button" onclick="wakeUp();">
          Wake Up!</button>
    </div>
</div><!-- end of MainContent -->
```

This button will show the text "Wake Up!" and call the "wakeUp()" function when clicked. Inside the wakeUp() function we simply display an alert() to show the seahorse is now awake.

Save your "seahorse.html" changes and reload it in your web browser to test the changes.

You should now see a new button near the bottom of the "MainContent" area.

Wake PopUpClicking on the button will show the alert pop-up.

Section 3. Managing Elements with JavaScript

Like most animals, the Seahorse gets grouchy if you try to poke him awake too many times. So let's add some code to our wakeUp() function to prevent the user from clicking on the button twice. We can also change the photo border to a bright yellow color to show the Seahorse is awake.

1. Run Komodo Edit and load your "seahorse.html" page.
2. Find the <img> element within the <figure> and add a new id attribute as shown below.

```
<figure>
  <img id="seahorseImg" src="PagePhotos/seahorse.jpg"
      alt="A seahorse." width="200" height="281">
  <figcaption>A Seahorse</figcaption>
</figure>
```

Now we can find this element when we want to from our JavaScript code.

3. Next, find your element surrounding the <button> and give it a unique id also.

```
<div id="wakeUpButtonDiv">
  <button type="button" onclick="wakeUp();">Wake Up!</button>
</div>
```

4. Find your wakeUp() function and add two new lines of code. The first will remove the button so it can't be clicked a second time, replacing it with some text. The second will change the <img> border color to yellow. Because each statement is rather long, we have shown them wrapped across two lines. But you can write each statement on a single line.

```
function wakeUp()
{
  // display an alert for the user
  alert("OK, OK, I'm awake.");
  document.getElementById("wakeUpButtonDiv").innerHTML=
                "<p><strong>I am Awake</strong></p>";
  document.getElementById("seahorseImg").style.borderColor="yellow";
}
```

5. Save your "seahorse.html" changes and reload it in your web browser to test the changes.

When you click on the button, you should see the same pop-up message. Then after you click "OK" to close the pop-up, the button should be replaced by text, and the image border should change to yellow.

Section 4. External JavaScript Files

1. You can use Komodo Edit to create a new JavaScript file for you. From within Komodo Edit, find the directory in your project folder such as such as "Scripts" where the new file will go. Right-click on that directory and select "New File from Template...". Scroll down to the "JavaScript" option and double-click on that line.
2. Type in your new filename such as "myScript.js", including the ".js" extension, and click "OK" to create your new file.
3. You should see your new file immediately in the project folder, and you can open it in the center editing area to start adding your JavaScript functions.

Section 5. Cuttle Fish Page

In this activity, you are going to create a new "Cuttlefish" page with several JavaScript features. A Cuttlefish is a shy creature, so users of this web page will see the following behavior:

- When the page loads, an alert pop-up will warn the user to be quiet.
- If the user hovers the mouse over the cuttlefish image, the picture will shrink as the cuttlefish tries to hide. Moving the mouse away from the image will restore it to normal size.
- If the user clicks on a "Boo!" button, the cuttlefish will hide completely and the button will change to read "Come back!". When clicked again, the image will re-appear and the button text will change back to "Boo!".

Let's begin by adding the Cuttlefish page and image. Find the Exercise16 folder in the zip file and copy Exercise16 folder to the Exercise folder on your computer. Copy the files from the folder (only the files, not the foler) into your "Aquamaniacs" folder.

Load "cuttlefish.html", which was copied to your Aquamaniac project early in your web browser to see the default look. A captioned photo and some text appear in the "MainContent" area. A button labeled "Boo!" has already been created, but will not do anything when you click on it.

Here are the important parts of the HTML code inside "cuttlefish.html".

```
<div id="MainContent">
<h1>The Cuttlefish</h1>
  <figure>
    <img id="cuttlefishImg" src="PagePhotos/cuttlefish.jpg"
              alt="A cuttlefish." width="250" height="281">
    <figcaption>A Cuttlefish</figcaption>
  </figure>
  <p>The cuttlefish is an amazing creature... </p>
    <div id="buttonDiv">
    <button type="button">Boo!</button>
  </div>
</div><!-- end of MainContent -->
```

You can see the <img> element has a unique id "cuttlefishImg" and the around the <button> has an id equal to "buttonDiv". None of the JavaScript functions have been written yet, and the elements do not yet have the event attributes set up to call any functions.

Step 1 – Display a Warning when the Page Loads

To display a warning when the page loads, you can simply add a <script> element right before the end of the </body>.

```
<script>

</script>
</body>
```

Any JavaScript statements you place within the body will run as the page loads, unless you have them wrapped up in a function.

1. Add the <script> element at the end of the <body> as shown above.
2. Inside the <script>, add an alert() statement with this text:
   "Shhh! If you spook the cuttlefish, he may hide!"
3. Save your "cuttlefish.html" changes and load the page in your web browser to check the results.

   You should see a pop-up display as soon as the page loads:

Step 2 – Shrink and Grow with MouseOver / MouseOut

Next, we want to make the cuttlefish image shrink when the mouse is over it and expand back to normal size when the mouse moves away. We will do this by creating two JavaScript functions, mouseOver() and mouseOut().

1. Find the <img> element with an id of "cuttlefishImg"
   - Add a onmouseover event attribute to call the mouseOver() JavaScript function
   - Add a onmouseout event attribute to call the mouseOut() JavaScript function.
2. We are going to store the JavaScript functions in a separate file called "Scripts/cuttlefish.js". So add a link to that file now with a new <script> element at the bottom of the page:

   ```
   </script>
   <script src="Scripts/cuttlefish.js"></script>
   </body>
   ```
3. Using Windows Explorer, Mac OS Finder, or Komodo Edit, create a new directory called "Scripts" inside your "Aquamaniacs" folder.
4. From Komodo Edit, right-click on the new "Scripts" folder and select "New File from Template...".
   - Double-click on the "JavaScript" line.
   - Type in "cuttlefish.js" for your new filename.
   - Click "OK" when finished to create your new external JavaScript file.
5. In your new "cuttlefish.js" file, add two new functions called mouseOver() and mouseOut(). Remember, in an external JavaScript file, you do not need to add the <script> element.

   ```
   function mouseOver()
   {
   }

   function mouseOut()
   {
   }
   ```
6. Add JavaScript statements into the mouseOver() function to get the "cuttlefishImg" element by id, and set the following styles:
   - Set width to "100px"
   - Set height to "100px"
7. Add JavaScript statements into the mouseOut() function to get the "cuttlefishImg" element by id, and set the following styles:
   - Set width to "" (an empty string with just opening and closing quotes)
   - Set height to ""

   An empty string with just opening and closing quotes will reset the property to its initial setting. So, this will return the image to its original size!

8. Save your "cuttlefish.html" and "cuttlefish.js" files and reload "cuttlefish.html" in a web browser to check your work. When you mouse over the cuttlefish image, it should shrink to a smaller size.

   Moving your mouse away from the image should return it to a normal size.

Step 3 – Hide and Show with Button Clicks

In this step, we want to add an onclick event function to the "Boo!" button in the HTML file.

1. Go ahead and add the onclick attribute now and make that event call the "hide()" function.

```
<div id="buttonDiv">
  <button onclick="hide();" type="button">Boo!</button>
</div>
```

2. Now, in your "cuttlefish.js" file, scroll down to the bottom and add new functions: hide() and show().
3. Inside the hide() function, add two JavaScript statements.
   - Get the "cuttlefishImg" element by id, and set the display property equal to "none". This will hide the cuttlefish completely when the button is clicked.
   - Get the "buttonDiv" element by id, and set the innerHTML property equal to:

     "<button type='button' onclick='show();'>Come back!</button>";

     This will create a button with "Come back!" display text that calls the show() function when clicked.

4. Inside the show() function, add two JavaScript statements.
   - Get the "cuttlefishImg" element by id, and set the display property equal to "block". This will restore (show) the cuttlefish when the button is clicked.
   - Get the "buttonDiv" element by id, and set the innerHTML property equal to:

     "<button type='button' onclick='hide();'>Boo!</button>";

     This will create a button with "Boo!" display text that calls the hide() function when clicked.

5. Save your "cuttlefish.js" changes and reload "cuttlefish.html" in your web browser to test the changes.

Now when you click on the "Boo!" button, the cuttlefish should disappear and the button text will change to "Come back!"

Clicking on the "Come back!" button should restore the page to normal and show the "Boo!" button again.