Interactive Web Computing
Assignment 1
Antony Magee
B00447842

Week 1: Wed 12th

1.1 Accordions

Some of the entries in these early work packages were completed at a later date as I began to tie the whole website together, they are mostly Cascading Style Sheets(CSS) and formatting additions to fit in with the rest of the site.

I began the assignment by taking the accordion demo from the practical lecture and playing around with it to allow it to fill the instead of being restricted in size, this was easily done by changing the CSS for the .accordion from a statics margin to, `width: auto;`. I then played around with the text size and color of the titles and the <p>'s of the text, the original text of the accordion was replaced with information about famous scientists, I wanted to add a picture to each of the entries so I searched google images by icon and displayed the pixel size of the images to find an image for each scientist that would fit well below the text. When adding the images I noticed that the CSS for the images was set to left justification so I added 2 lines to center it, margin-left: auto; , margin-right: auto; . When adding the images I gave them a title to be able to refer to them with jQuery at a later date,

```
<img title="Richard Feynman" src="richard.jpg" />
```

This allowed me to use the title as data to be captured by the sprite/tool tip mouse over function, when mouseing over the images would display the name of the scientist in the tool tip. Below the image I added a link for further reading to take the user to the wikipedia page to the scientist they were viewing. I had some initial problems with this as the link would just open the page in the current tab and write over the assignment web page so I did some research and added the

```
target="_blank"
```

To the link so that it would open a new tab and not interrupt the browsing of the current tab. To be able to align these under the images I added tags before and after the links to make them display below each other.

The original border was a dotted one and it didn't fit in well with the other CSS is was using for the theme of the site I changed it to be a 1px solid border with the same color as the rest of my boarders. The images were also wrapped in this CSS to give them the same border with slightly different settings but to the same effect.

1.1.1 *jQuery*

The jQuery I used was simple, in that it just opened the tab you clicked, and does not dynamically hide all other open divs but just opens additional and you have to click the open div again to close it.

```
$(document).ready(function()  {
       $('.scientist_name').click(function()
       {
        $(this).next().slideToggle(200);
       });
});
```

When clicking a paragraph with the class scientist_name it slides it with the speed of 200, this then exposes the body of the accordion tab. It is set up so that all the visible names are <p> and the body associated with each is a named scientist_description.

1.1.2 *CSS*

For the main body of the accordion I did not want any margins or borders to mess with the clean look of it within the parent . Each of the scientists names were given a little padding and a background color, the same color was used for hover highlighting in other parts of the website. There is a 1px margin to allow the user to see the separation between the accordion tabs, and the cursor is instructed to become a pointer when hovering over the names to indicated there ability to be clicked.

```
.scientist_name {
  padding: 6px 10px;
  cursor: pointer;
  background-color:#999966;
  margin:1px;
  font-size: 100%;  }
```

```
.scientist_description {
  display: none;
  padding: 5px 5px 5px 5px;
  font-size: 90%;
  border: 1px solid #ddd;  }
```

For the scientist_description there was an area of padding set to indent all the text and the font size was reduced to make it more fitting with being further information about the above header.

At the top of this page the header is nicely placed with a good area of white space to make it highly visible and eye catching this was done by specifying that all <h2> that were .tab_contents would have a set style, <h2> already creates it's own white space above it so I only needed

to add some below to give it a little room. Then a border was added to define the separation between page title and content.

```
.tab_content h2 {
font-weight: normal;
padding-bottom: 10px;
border-bottom: 1px dashed #ddd;
font-size: 1.8em;  }
```

Week 2: Sat-Sun 15th-16th:

1.2 Hide/Show/Fade

For my hide, show and fade demo I wanted to make it slightly interactive, so thinking along those lines and knowing that I wanted to use the Click event handler I settled on making a simple whack-a-mole game. Again I used google search to find an image of the correct size, then I created a table with id mole_table to hold all the images, I also had to store the title to be used with the tool tip and an id so that I could identify them when triggering the Click function. I also had them in a with an ID for referencing.

```
<td>
	<div id="mole1">
		<img title="The first little mole"  id="img1" src="mole.jpg" />
	</div>
</td>

<table id="mole_table" border="0">
```

The border of the table was made 0 so all the user could see is the 6 images floating nicely in the center of the screen.
After the table was created with the 6 images in it and tags placed around it I could add the button that would reset the game and make the images visible again.

```
<input type="button" id="reset_button" value="Reset Game">
```

1.2.1 *jQuery*

This was basically the page setup and all I had to do now was add the functionality. To hide the mole images when clicked I added all the move id's to a function all and used the *this* option to be able to determine which div it was clicking. The slow paramater was to set the hide animation so that it would not be instant.

```
$('#mole1, #mole2, #mole3, #mole4, #mole5, #mole6').click(function() {
			$(this).hide('slow');
	});
```

When clicked the button now smartly hides the selected div until they are all gone, and because the parent div is set to auto side it reduces it's size correctly. To be able to reset the 's back to there original visible state I created another button to carry out this functionality. Here I just added all the 's to the button and attached a .show() call.

```
$('#reset_button').click(function(event) {
	$("#mole1, #mole2, #mole3, #mole4, #mole5, #mole6").show();
});
```

I wanted to allow the user to control the opacity of the 's so I added 2 more buttons to both increase and reduce the opacity between 0.33 and 1.

```
$('#fade_button').click(function(event) {
	$("#mole1, #mole2, #mole3, #mole4, #mole5, #mole6").fadeTo("slow",   0.33);
	});

$('#unfade_button').click(function(event) {
	$("#mole1,#mole2, #mole3, #mole4, #mole5, #mole6").fadeTo("slow", 1);
	});
```


The .hide() and .show() calls both still work when the images are faded.

2.1.2 *CSS*

The CSS for this section was mostly taken from standard stuff, such as the image borders and container div settings, most of the work on this page was jQuery and after the initial setup of the images and buttons very little was done visually.

Week 2: Wed 19th

2.2 Tabs

This was implemented after a good bit of research around various CSS and web development sites, I learned that you initially had to create a unordered list with all the titles of the tabs that

you wanted to create so I made one and popped it in below my <h1> tag within my .container .

```
<ul class="tabs">
	<li><a href="#tab0">Welcome</a></li>
</ul>
```

This was the easy bit, to create the headings to be used by the jQuery code to implement the tabs, the result is a simply row of options that open the corresponding it does not look nice and will need a lot of CSS work to make it presentable.

2.2.1 *jQuery*

So to start I wanted to hide all the contents of the tabs, then activate the first tab in the list, in this case the Home tab with come basic info on and then display the corresponding div that went with the tab title.

```
$(".tab_contents").hide();
	$("ul.tabs li:first").addClass("active").show();
	$(".tab_contents:first").show();
```

That sorted out the initial display of the tabs but to get them to work with the click(); event some more work was needed. I placed a click(); listener on the tabs and instructed it when clicked to remove any active classes, add the active content to the tab, then hide() all the tab content, then find the correct attribute to identify the active content and then fadeIn() the correct content to the selected tab.

```
$("ul.tabs li").click(function() {
	$("ul.tabs li").removeClass("active");
	$(this).addClass("active");
	$(".tab_contents").hide();
	var activeTab = $(this).find("a").attr("href");
	$(activeTab).fadeIn();
	});
```

This then allowed the correct content to be displayed when the corresponding tab was clicked, but it still looked pretty terrible form a aesthetic standpoint so I had to add quite a lot of CSS to make it presentable.

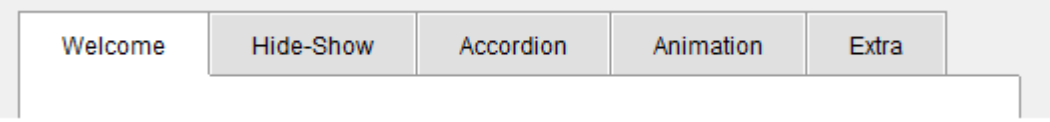
2.2.2 *CSS*

For this section I used a lot of CSS to make all the elements visibly pleasing and easy to use, firstly I skinned the <ul> tabs and the and <li> and <a> elements within it.

```
ul.tabs {
	margin: 0;
	padding: 0;
	float: left;
	list-style: none;
	height: 32px;
	border-bottom: 1px solid #999;
	border-left: 1px solid #999;
	width: 100%;
}
```

```
ul.tabs li {
	float: left;
	margin: 0;
	padding: 0;
	height: 31px;
	line-height: 31px;
	border: 1px solid #999;
	border-left: none;
	margin-bottom: -1px;
	background: #e0e0e0;
	overflow: hidden;
	position: relative;
}
```

```
ul.tabs li a {
	text-decoration: none;
	color: #000;
	display: block;
	font-size: 1.2em;
	padding: 0 20px;
	border: 1px solid #fff;
	outline: none;
}
```

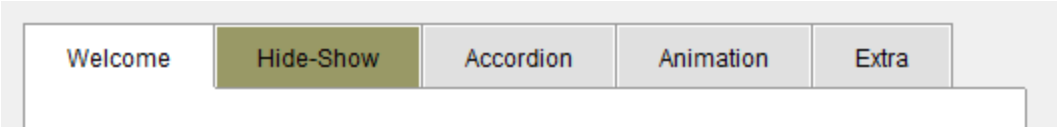


This made the tabs at the top of the page a lot better looking, to add the hover functionality while the mouse was over a tab I needed to add a hover listener which can be added directly within CSS so here are the 2 pieces of CSS to deal with shading and hovering the selected tab headings.

```
ul.tabs li a:hover {
	background: #999966;
}
```

```
html ul.tabs li.active, html ul.tabs
li.active a:hover  {
	background: #fff;
	border-bottom: 1px solid #fff;
}
```

The color chosen for the hover function was the same as the accordion, to try and keep some continuity between pages.



Week 3: Wed 26th

3.1 Animation

This week started with the animation and deciding what to do for it, while looking around on the jQuery site I saw a demo where you could move a colored from left to right with the animate() function. I wanted to use this to create an interactive game so I sourced a maze image from google and placed it on the animation page and found an image of a little robber to run around the maze to his home.

The fist problem was positioning the robber image on the maze as window re-sizes would move his position, I had to research relative positioning and place him in relation to the parent

```
.tab_contents {
 position:relative;
}
```

```
<div class="thief_div"
       style="
    position:absolute;
       top:153px;
       left:260px;
       z-index: 1;
       visibility: show;">

       <img src="thief.png" />
```

I added the style for the div within it as it was giving problems when placed with the rest of the CSS. After the images were placed correctly i placed 4 buttons at the bottom to control the direction the robber could move around the maze.

```
<form>
               <input type="button" id="up_button" value="Up">
               <br></br>
               <input type="button" id="left_button" value="Left">
               <input type="button" id="right_button" value="Right">
               <br></br>
               <input type="button" id="down_button" value="Down">
       </form>
```

3.1.1 *jQuery*

The animation for this page is simply attaching the animate() function to the thief and telling it to move the image in relation to the top and left page sides.

```
$("#right_button").click(function(event){
               $(".thief_div").animate({"left": "+=50px"}, "slow");
       });

       $("#left_button").click(function(event){
               $(".thief_div").animate({"left": "-=50px"}, "slow");
       });
```

```
$("#up_button").click(function(event){
	$(".thief_div").animate({"top": "-=50px"}, "slow");
});

$("#down_button").click(function(event){
	$(".thief_div").animate({"top": "+=50px"}, "slow");
});
```

3.2 Sprite/Tooltip

This only needed basic changes from the practical the make it viable for my website, mainly changing the e in the functions to event and adding <title> tags to elements I wanted to be displayed in the tool tip. I added this to all images to add some extra mouse over information.

```
<img title="Sir Isacc Newton" src="newton.png" />

<img title="The first little mole"  id="img1" src="mole.jpg" />
```

3.2.1 *jQuery*

After adding the titles to the elements I wanted to display it was just a matter of changing the practical code to allow me to grab the title of the image and pass it to the tool tip.

```
function sprite(){
	xOffset = 10;
	yOffset = 20;


		$("img").hover(
			function(event){
			this.t = this.title;
			this.title = "";
				$("body").append("<p id='sprite'>" + this.t +"</p>");

		  $("#sprite")
		    .css("top",(event.pageY - xOffset) + "px")
		    .css("left",(event.pageX + yOffset) + "px");
		    },

		  function(){
			this.title = this.t;
			$("#sprite").remove();
			});

		  $("img").mousemove(function(event){
		   $("#sprite")
		    .css("top",(event.pageY - xOffset) + "px")
```

```
                .css("left",(event.pageX + yOffset) + "px");
                   });

        };
```

3.3 Dynamically created Drop down List

I initially had some trouble creating this as it was hard to include the id of the list item while dynamically creating it, this was overcome with some help and a little fiddling with double and single quotes, firstly I created a span with elements that were to be passed to jQuery to create the list.

```
<div id="opacity_options">
    <span>Mole 1</span> <span>Mole 2</span> <span>Mole 3</span> <span>Mole 4</span>
    <span>Mole 5</span> <span>Mole 6</span>
  </div>
```

Once this was done I was able to pass the information onto the jQuery to handle the creation of the select. And the where the text is to be output is right beneath the containing the drop down list.

```
<div id="opacity_drop"></div><br />

<div class="mole_select"></div><br />
```

3.3.1 *jQuery*

```
$('#opacity_drop').append('<select>');
                        var i = 0;
$('span').each(function(){

$('select').append('<option value="' + $(this).text() +   '" id="' + i + '">' + $(this).text()
+ '<\/option>' );
                         i += 1;
});
$("select").change(function () {
var str = "";
$("select option:selected").each(function () {
      str += $(this).text() + " ";
      });
$(".mole_select").text(str).append("Has been chosen, he is very happy you chose          him,
he dosen't get out much!");
        })
```

These snippets handle the reading in of the elements and the adding of the id tags as it does so. The the change() function catches when an option is picked and the text of the selection is stored in a string then output with some extra text .append() to it.

Overall CSS

Some CSS was not used in any of the given examples but was used around the site such as <h2> tags and div formatting. I will go through them in this section.

```
.tab_contents {
	padding: 20px;
	font-size: 1.2em;
}


.tab_contents p {
	font-weight: normal;
	font-size: 1.3em;
}

.tab_container {
	border: 1px solid #999;
	border-top: none;
	clear: both;
	float: left;
	width: 100%;
	background: #fff;
	-moz-border-radius-bottomright: 5px;
	-khtml-border-radius-bottomright: 5px;
	-webkit-border-bottom-right-radius:
5px;
	-moz-border-radius-bottomleft: 5px;
	-khtml-border-radius-bottomleft: 5px;
	-webkit-border-bottom-left-radius:
5px;
}
```

```
.tab_contents h2 {
	font-weight: normal;
	padding-bottom: 10px;
	border-bottom: 1px solid #ddd;
	font-size: 1.8em;
}

.tab_contents h3 {
	font-weight: normal;
	font-size: 1.8em;
}

.tab_contents img {
	float: center;
	margin: 0 2px 2px 0;
	border: 1px solid #ddd;
	padding: 2px;
}


img {
	cursor: pointer;
}
```