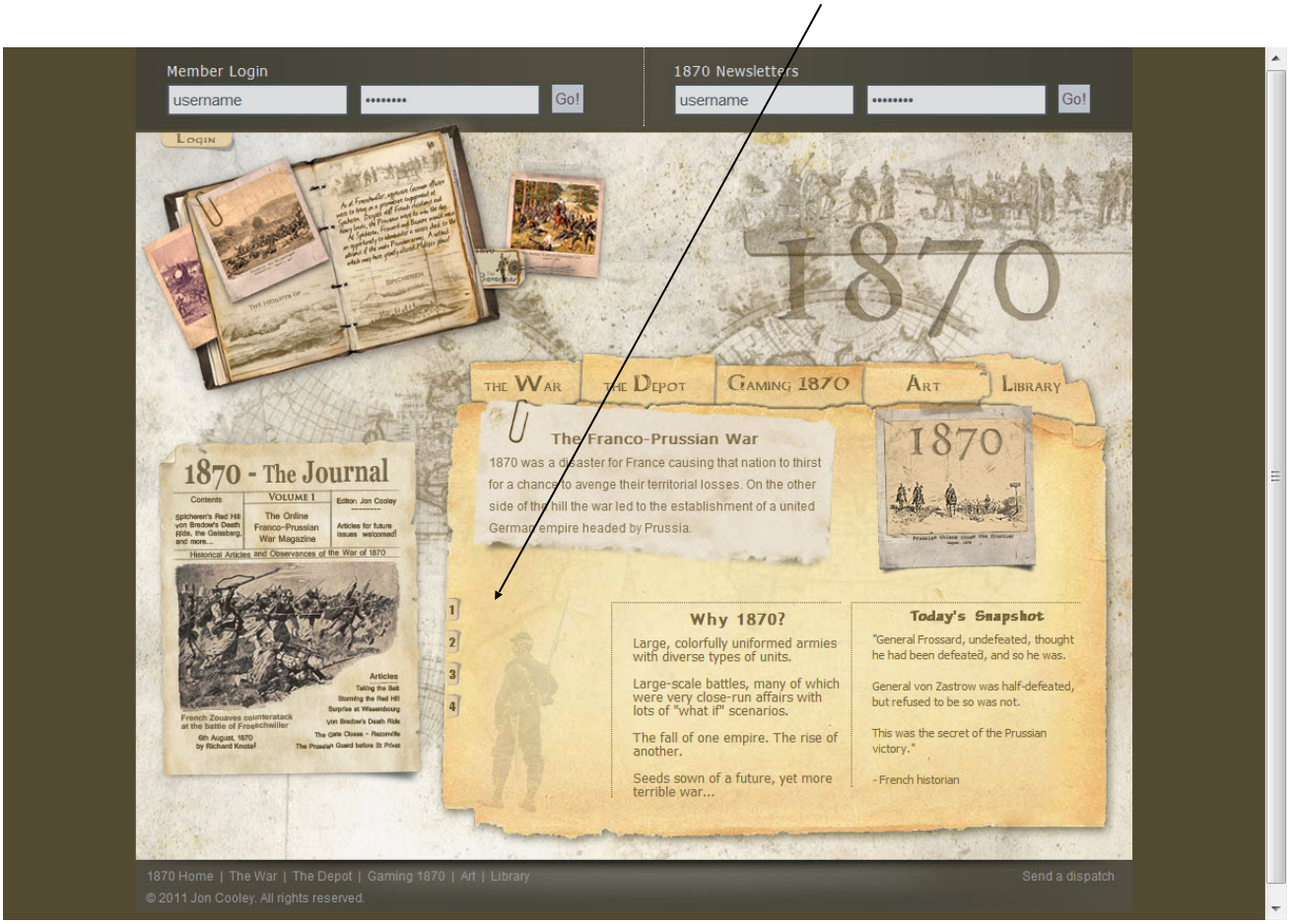
# Web 2 - JavaScript Events Assignment

Using the provided **fpwClickEvent_startFiles.zip** folder to begin this assignment. You will have a web page named **fpwStage03.html** along with three folders (***css***, ***images***, and ***js***).

1. Update the **1870primary.css** file (in the ***css*** folder) starting on line 410 selecting the ** ** and adding styles to position the div *absolutely* so the numbered tab images appear as shown below:




Also, add a CSS transition attribute to the **<aside class="picText">** on line 386 that transitions any change to this element's opacity property to take 2 seconds to occur with a default CSS ease value applied.

2. Create an external JavaScript file under the ***js*** folder named **imagePicker_yourname.js** (note: this naming convention will be graded to see if you are following the specs). In this JavaScript file do the following:

   a) Define an object literal named **ImagePicker** which will contain your code and add an ***onload*** event handler attribute to the <body> tag in the .html file to run the **ImagePicker** object's **init()** method when the page's content finishes loading. This and updating the <script> tag to include your properly named .js file should be the only changes you need to make in the .html file.

   b) Above the **init()** method of your **ImagePicker** object, but inside the object literal, use object literal syntax to declare and initialize the following class-level property variables that either contain a reference to a tag or contain a plain value:

- to the <aside> tag that has a class of "***picText***" storing it in a class-level property of ImagePicker named **imageAside**
- to the image having an id of "***pic***" storing it in a class-level property of ImagePicker named **imagePic**
- Set up a class-level property of ImagePicker named **oldImagePicId** storing an initial value of 0 in it
- Set up a class-level property of ImagePicker named **newImagePicId** storing an initial value of 0 in it

c) Inside the init() method, get a reference to ** ** storing it in a *local variable* named **imageTabsDiv**.

d) Next, get references to all <img> tags in div#imageTabs storing them in a local node list variable named **imageTabsList**.

e) Step through the **imageTabsList** node list one element at a time using a **for loop** and assign a **click event handler** method called **imageTabClick()** to each element in the node list.

Use the older event handler format: ***elementReference.onevent*** = ***handlerFunctionName;*** (note: since the ***handlerFunctionName*** is a method of our object, be sure to reference it as ***objectName.methodName***).

f) Still in the **init()** method, write some code to randomly pick one of the four images to display initially once the page is loaded (without hardcoding). The paths to these images are the values of the image tab element's ***longdesc*** attribute in the .html file.

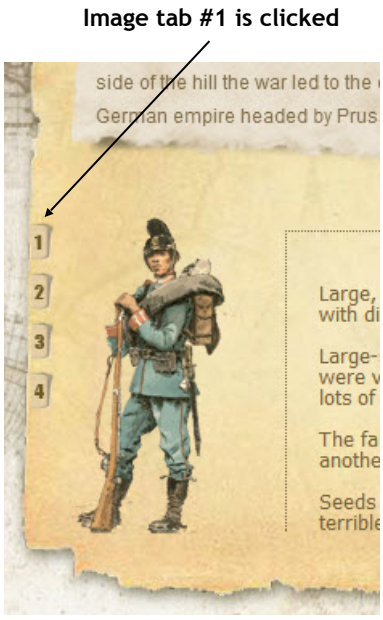
Hint: generate a random number (store in local variable named **randomIndex**) to use as an index into the **imageTabsList** node list to get the image tab element's ***longdesc*** attribute's value for use as the image's source file.

Also, reset the *newImagePicId* property's value to the random image tab element's id value.

g) In the **imageTabClick()** method (function) definition:

- Swap the values of the *oldImagePicId* and *newImagePicId* properties where the *oldImagePicId* needs to be replaced by the *newImagePicId* and the *newImagePicId* should be replaced by the clicked image tab's id value.
- In one single statement, select the ***paragraph tag*** inside aside#picText whose id is "p" followed by the value of *oldImagePicId* and set its CSS **display** attribute to a value of "**none**".
- use the ***this*** keyword to reference the tab image that was clicked and call its **getAttribute("attributeName")** method to get the path ***to the image associated with that image tab*** storing it in a variable named **imgPath** (hint: the ***longdesc*** attribute for the tab image contains this path in the .html file).

- Use the **imgPath** variable to perform the image rollover (on the **imagePic** image property that you created a reference to above in **part b**) to the image associated with the image tab that was clicked.

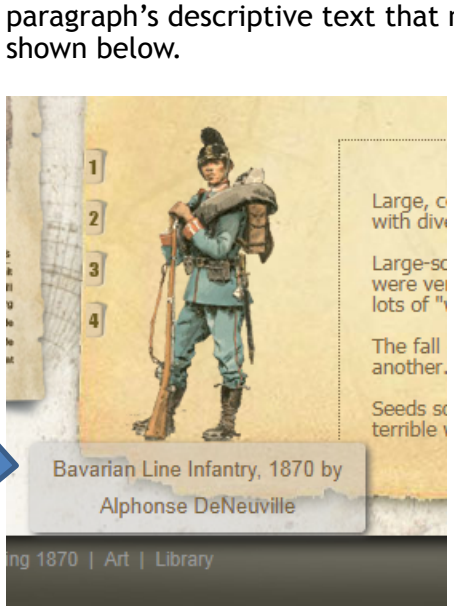



h) Now, in the init() method, create a mouseenter event listener (new way) on the imagePic property's image that calls an event handler method of your object named **showImageDescription**.

This event handler's code should perform the following tasks:

- make the **<aside class="picText">** tag fully opaque by setting its *opacity* style
- In one statement, select the <p> tag whose id value matches "p" plus the *newImagePicId*'s integer value **and** show the paragraph via its **display** CSS attribute.

This will cause the <aside> tag's box to appear using a CSS transition to fade it in showing the paragraph's descriptive text that matches the *imagePic's* image currently being displayed as shown below.



i) Finally, in the init() method, create a **mouseleave** event listener (new way) on the *imagePic* property's image that calls an event handler method of your object named **hideImageDescription**.

This event handler's code should make the **<aside class="picText">** tag transparent by setting its *opacity* style.

This will cause the <aside> tag's box to fade out over a short time until it is no longer visible.

3. Zip up your files into a zipped folder named **fpwClickEvent_yourname.zip** and submit via the Dropbox for this assignment.