This assignment is due before lesson 5, to BCIT’s “Desire to Learn” “assignment1” dropbox located at <http://learn.bcit.ca>

It is worth 10% of your overall grade for this course.

Do not use jQuery for this assignment. Do not add any extra DIVs etc to this assignment either.

Submit two files:

lastname\_firstname.html and lastname\_firstname.js (use your own first and last names).

Write HTML and JavaScript that behaves in the following way.

The HTML page contains three divs.

The top div has an ID of “top” and a class of “change”.

The middle div has an ID of “middle”.

The bottom div has an ID of “bottom” and a class of “change”.

Each of these three divs contains three images, three hyperlinks, and three paragraphs.

After the window is finished loading, use JavaScript to do the following:

Each of the paragraphs in the divs with class “change” will turn their font color to red when the mouse is on it, and change it to black when the mouse is off it.

Any paragraph on the page gets a black dashed 1px wide border when it gets clicked; the border goes away when clicked again; in other words, the border gets toggled on or off each time it is clicked.

Any image in a div whose ID is “top” or “middle” gets its display style set to “none” when it is double clicked.

Use DOM 1 methods getElementsByTagName() and getElementById() for this assignment.

Important: use object detection at least once in this assignment.

The bottom div’s links should not work if they contain the word “nhl” in their href attribute. (e.g. a link to “http://www.pgatour.com” would work, but a link to “http://www.nhl.com” would not). Make sure “http://www.nhl.com” is one of the links in each div.

## Important!

Copy/paste all of the contents from   
**Share Out/COMP/2015/homework/assignment1/assignment-1-unit-test.html**   
into your HTML file, just before your </body> tag. That script will run and give you a score out of 72. That will be your score for this assignment. The marking/testing script code is also listed on the following pages for your own interest. IMPORTANT: the test only runs properly in Chrome; use that for testing. Do not change the tests at all, and leave this code in your file when you submit it.

<link rel="stylesheet" href="https://code.jquery.com/qunit/qunit-1.23.1.css">

<script src="https://code.jquery.com/qunit/qunit-1.23.1.js"></script>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.2/jquery.min.js"></script>

<div id="qunit"></div>

<div id="qunit-fixture"></div>

<script>

function getRedString(color){

if((color == "rgb(255, 0, 0)") ||

(color == "red")){

//alert('red');

return "red";

}else{

return color;//"not red"

}

}

function getBlackString(color){

if((color == "rgb(0, 0, 0)") ||

(color == "black")){

//alert('black');

return "black";

}else{

return color; //"not black"

}

}

QUnit.test( "assignment 1 test", function( assert ) {

/\* test: The HTML page contains three divs.

The top div has an ID of "top" and a class of "change".

The middle div has an ID of "middle".

The bottom div has an ID of "bottom" and a class of "change".

\*/

assert.equal($('div:first').attr('id'), "top"); // test 1

assert.equal($('div:nth-of-type(2)').attr('id'), "middle"); // test 2

assert.equal($('div:nth-of-type(3)').attr('id'), "bottom"); // test 3

assert.equal($('div:first').attr('class'), "change"); // test 4

assert.equal($('div:nth-of-type(3)').attr('class'), "change"); // test 5

/\* test: Each of these three divs contains three images, three hyperlinks, and three paragraphs.

\*/

assert.equal($('div:first').find('img').length, 3); // test 6

assert.equal($('div#middle').find('img').length, 3); // test 7

assert.equal($('div:nth-of-type(3)').find('img').length, 3); // test 8

assert.equal($('div:first').find('a').length, 3); // test 9

assert.ok($('div:nth-of-type(2)').find('a').length >= 3); // test 10

assert.equal($('div:nth-of-type(3)').find('a').length, 3); // test 11

assert.equal($('div:first').find('p').length, 3); // test 12

assert.equal($('div:nth-of-type(2)').find('p').length, 3); // test 13

assert.equal($('div:nth-of-type(3)').find('p').length, 3); // test 14

/\* test: All of the paragraphs in the divs with class "change" will turn their font color to red when the

mouse is on them, and change it to black when the mouse is off them.

\*/

//$('div#top').find('p').first().trigger('mouseover');

//assert.equal($('div#top').find('p').first().css('color'), ("rgb(255, 0, 0)" || 'red')); // test 15

//alert($('div#top p:nth-of-type(2)').attr('id'));

/\* test: turns red on mouse over: top-div paragraphs

\*/

$('div#top p:first').trigger('mouseover');

assert.equal(getRedString($('div#top p:first').css('color')), 'red'); // test 15

$('div#top p:nth-of-type(2)').trigger('mouseover');

assert.equal(getRedString($('div#top p:nth-of-type(2)').css('color')), 'red'); // test 16

$('div#top p:nth-of-type(3)').trigger('mouseover');

assert.equal(getRedString($('div#top p:nth-of-type(3)').css('color')), 'red'); // test 17

/\* test: turns red on mouse over: bottom-div paragraphs

\*/

$('div#bottom p:first').trigger('mouseover');

assert.equal(getRedString($('div#bottom p:first').css('color')), 'red'); // test 18

$('div#bottom p:nth-of-type(2)').trigger('mouseover');

assert.equal(getRedString($('div#bottom p:nth-of-type(2)').css('color')), 'red'); // test 19

$('div#bottom p:nth-of-type(3)').trigger('mouseover');

assert.equal(getRedString($('div#bottom p:nth-of-type(3)').css('color')), 'red'); // test 20

/\* test: turns black on mouse out: top-div paragraphs

\*/

$('div#top p:first').trigger('mouseout');

assert.equal(getBlackString($('div#top p:first').css('color')), 'black'); // test 21

$('div#top p:nth-of-type(2)').trigger('mouseout');

assert.equal(getBlackString($('div#top p:nth-of-type(2)').css('color')), 'black'); // test 22

$('div#top p:nth-of-type(3)').trigger('mouseout');

assert.equal(getBlackString($('div#top p:nth-of-type(3)').css('color')), 'black'); // test 23

/\* test: turns black on mouse out: bottom-div paragraphs

\*/

$('div#bottom p:first').trigger('mouseout');

assert.equal(getBlackString($('div#bottom p:first').css('color')), 'black'); // test 24

$('div#bottom p:nth-of-type(2)').trigger('mouseout');

assert.equal(getBlackString($('div#bottom p:nth-of-type(2)').css('color')), 'black'); // test 25

$('div#bottom p:nth-of-type(3)').trigger('mouseout');

assert.equal(getBlackString($('div#bottom p:nth-of-type(3)').css('color')), 'black'); // test 26

/\* test: Any paragraph on the page gets a black dashed 1px wide border when it gets clicked;

the border goes away when clicked again; in other words, the border gets toggled on

or off each time it is clicked.

\*/

// top div's first paragraph...initially, not dashed

assert.equal(getDashedString($('div#top p:first').css('border-style')), 'not-dashed'); // test 27

// dashed 1px black border toggles with each click

$('div#top p:first').trigger('click');

assert.equal(getDashedString($('div#top p:first').css('border-style')), 'dashed'); // test 28

assert.equal(getBlackString($('div#top p:first').css('border-color')), 'black'); // test 29

assert.equal($('div#top p:first').css('border-width'), '1px'); // test 30

$('div#top p:first').trigger('click');

assert.equal(getDashedString($('div#top p:first').css('border-style')), 'not-dashed'); // test 31

$('div#top p:first').trigger('click');

assert.equal(getDashedString($('div#top p:first').css('border-style')), 'dashed'); // test 32

// top div's second paragraph...initially, not dashed

assert.equal(getDashedString($('div#top p:nth-of-type(2)').css('border-style')), 'not-dashed'); // test 33

// dashed 1px black border toggles with each click

$('div#top p:nth-of-type(2)').trigger('click');

assert.equal(getDashedString($('div#top p:nth-of-type(2)').css('border-style')), 'dashed'); // test 34

assert.equal(getBlackString($('div#top p:nth-of-type(2)').css('border-color')), 'black'); // test 35

assert.equal($('div#top p:nth-of-type(2)').css('border-width'), '1px'); // test 36

$('div#top p:nth-of-type(2)').trigger('click');

assert.equal(getDashedString($('div#top p:nth-of-type(2)').css('border-style')), 'not-dashed'); // test 37

$('div#top p:nth-of-type(2)').trigger('click');

assert.equal(getDashedString($('div#top p:nth-of-type(2)').css('border-style')), 'dashed'); // test 38

// top div's third paragraph...initially, not dashed

assert.equal(getDashedString($('div#top p:nth-of-type(3)').css('border-style')), 'not-dashed'); // test 39

// dashed 1px black border toggles with each click

$('div#top p:nth-of-type(3)').trigger('click');

assert.equal(getDashedString($('div#top p:nth-of-type(3)').css('border-style')), 'dashed'); // test 40

assert.equal(getBlackString($('div#top p:nth-of-type(3)').css('border-color')), 'black'); // test 41

assert.equal($('div#top p:nth-of-type(3)').css('border-width'), '1px'); // test 42

$('div#top p:nth-of-type(3)').trigger('click');

assert.equal(getDashedString($('div#top p:nth-of-type(3)').css('border-style')), 'not-dashed'); // test 43

$('div#top p:nth-of-type(3)').trigger('click');

assert.equal(getDashedString($('div#top p:nth-of-type(3)').css('border-style')), 'dashed'); // test 44

// bottom div's first paragraph...initially, not dashed

assert.equal(getDashedString($('div#bottom p:first').css('border-style')), 'not-dashed'); // test 45

// dashed 1px black border toggles with each click

$('div#bottom p:first').trigger('click');

assert.equal(getDashedString($('div#bottom p:first').css('border-style')), 'dashed'); // test 46

assert.equal(getBlackString($('div#bottom p:first').css('border-color')), 'black'); // test 47

assert.equal($('div#bottom p:first').css('border-width'), '1px'); // test 48

$('div#bottom p:first').trigger('click');

assert.equal(getDashedString($('div#bottom p:first').css('border-style')), 'not-dashed'); // test 49

$('div#bottom p:first').trigger('click');

assert.equal(getDashedString($('div#bottom p:first').css('border-style')), 'dashed'); // test 50

// bottom div's second paragraph...initially, not dashed

assert.equal(getDashedString($('div#bottom p:nth-of-type(2)').css('border-style')), 'not-dashed'); // test 51

// dashed 1px black border toggles with each click

$('div#bottom p:nth-of-type(2)').trigger('click');

assert.equal(getDashedString($('div#bottom p:nth-of-type(2)').css('border-style')), 'dashed'); // test 52

assert.equal(getBlackString($('div#bottom p:nth-of-type(2)').css('border-color')), 'black'); // test 53

assert.equal($('div#bottom p:nth-of-type(2)').css('border-width'), '1px'); // test 54

$('div#bottom p:nth-of-type(2)').trigger('click');

assert.equal(getDashedString($('div#bottom p:nth-of-type(2)').css('border-style')), 'not-dashed'); // test 55

$('div#bottom p:nth-of-type(2)').trigger('click');

assert.equal(getDashedString($('div#bottom p:nth-of-type(2)').css('border-style')), 'dashed'); // test 56

// bottom div's third paragraph...initially, not dashed

assert.equal(getDashedString($('div#bottom p:nth-of-type(3)').css('border-style')), 'not-dashed'); // test 57

// dashed 1px black border toggles with each click

$('div#bottom p:nth-of-type(3)').trigger('click');

assert.equal(getDashedString($('div#bottom p:nth-of-type(3)').css('border-style')), 'dashed'); // test 58

assert.equal(getBlackString($('div#bottom p:nth-of-type(3)').css('border-color')), 'black'); // test 59

assert.equal($('div#bottom p:nth-of-type(3)').css('border-width'), '1px'); // test 60

$('div#bottom p:nth-of-type(3)').trigger('click');

assert.equal(getDashedString($('div#bottom p:nth-of-type(3)').css('border-style')), 'not-dashed'); // test 61

$('div#bottom p:nth-of-type(3)').trigger('click');

assert.equal(getDashedString($('div#bottom p:nth-of-type(3)').css('border-style')), 'dashed'); // test 62

/\* test: Any image in a div whose ID is "top" or "middle" gets its display style set to "none" when it is double

clicked.

\*/

// top div's first image:

$('div#top img:first').trigger('dblclick');

assert.equal($('div#top img:first').css('display'), 'none'); // test 63

// top div's second image:

$('div#top img:nth-of-type(2)').trigger('dblclick');

assert.equal($('div#top img:nth-of-type(2)').css('display'), 'none'); // test 64

// top div's third image:

$('div#top img:nth-of-type(3)').trigger('dblclick');

assert.equal($('div#top img:nth-of-type(3)').css('display'), 'none'); // test 65

// middle div's first image:

$('div#middle img:first').trigger('dblclick');

assert.equal($('div#middle img:first').css('display'), 'none'); // test 66

// middle div's second image:

$('div#middle img:nth-of-type(2)').trigger('dblclick');

assert.equal($('div#middle img:nth-of-type(2)').css('display'), 'none'); // test 67

// middle div's third image:

$('div#middle img:nth-of-type(3)').trigger('dblclick');

assert.equal($('div#middle img:nth-of-type(3)').css('display'), 'none'); // test 68

// bottom div's images must not react to double clicks

// bottom div's first image:

$('div#bottom img:first').trigger('dblclick');

assert.notEqual($('div#bottom img:first').css('display'), 'none'); // test 69

// bottom div's second image:

$('div#bottom img:nth-of-type(2)').trigger('dblclick');

assert.notEqual($('div#bottom img:nth-of-type(2)').css('display'), 'none'); // test 70

// bottom div's third image:

$('div#bottom img:nth-of-type(3)').trigger('dblclick');

assert.notEqual($('div#bottom img:nth-of-type(3)').css('display'), 'none'); // test 71

/\* test: Make sure "http://www.nhl.com" is one of the links in each div.

\*/

/\* test2: The bottom div’s links should not work if they contain the word "nhl" in their href attribute.

(e.g. a link to "http://www.pgatour.com" would work, but a link to "http://www.nhl.com" would not).

\*/

assert.equal(matchNHLInLink(), true); // test 72

});

function matchNHLInLink(div){

var foundNHLLink = false;

// look in top div for nhl.com link

if($('div#top a:first').attr('href').indexOf("nhl.com") != -1){

foundNHLLink = true;

}else if($('div#top a:nth-of-type(2)').attr('href').indexOf("nhl.com") != -1){

foundNHLLink = true;

}else if($('div#top a:nth-of-type(3)').attr('href').indexOf("nhl.com") != -1){

foundNHLLink = true;

}

if(foundNHLLink == false){

// top div doesn't have NHL link

return false;

}else{

// reset and look in second div

foundNHLLink = false;

}

// look in middle div for nhl.com link

if($('div#middle a:first').attr('href').indexOf("nhl.com") != -1){

foundNHLLink = true;

}else if($('div#middle a:nth-of-type(2)').attr('href').indexOf("nhl.com") != -1){

foundNHLLink = true;

}else if($('div#middle a:nth-of-type(3)').attr('href').indexOf("nhl.com") != -1){

foundNHLLink = true;

}

if(foundNHLLink == false){

// middle div doesn't have NHL link

return false;

}else{

// reset and look in third div

foundNHLLink = false;

}

// look in bottom div for nhl.com link

if($('div#bottom a:first').attr('href').indexOf("nhl.com") != -1){

foundNHLLink = true;

// trigger a click

//$('div#bottom a:first').click();

}else if($('div#bottom a:nth-of-type(2)').attr('href').indexOf("nhl.com") != -1){

foundNHLLink = true;

//alert('bottom div, second a, href is: ' + $('div#bottom a:nth-of-type(2)').attr('href'));

// trigger a click

$('div#bottom a:nth-of-type(2)').triggerHandler('click');

}else if($('div#bottom a:nth-of-type(3)').attr('href').indexOf("nhl.com") != -1){

foundNHLLink = true;

// trigger a click

//$('div#bottom a:nth-of-type(3)').click();

}

if(foundNHLLink == false){

// bottom div doesn't have NHL link

return false;

}else{

// it has been found

return true;

}

}

function getDashedString(string){

if(string != "dashed"){

return "not-dashed";

}else{

return "dashed";

}

//alert(string);

}

</script>