As part of the redesign of the library website, we wanted to automatically display the current day’s open hours. But I only wanted to do it if I could find a way to automatically change the hours on holidays. These scripts compare the current date with a list of holidays, break periods, and other exceptional periods to control whether a special message should be displayed, and shows a message based on the day of the week when no special hours are in effect.

Files used in this demonstration can be found at:
   http://www2.leeward.hawaii.edu/library/todayshoursdemo.html
   http://www2.leeward.hawaii.edu/library/css/todayshoursdemo.css
   http://www2.leeward.hawaii.edu/library/js/libraryhours.js (the actual working file)

To learn more about HTML, JavaScript, & CSS: http://www.w3schools.com

**HTML Coding**

The spot in a web page where a script does what it does can be designated using class or ID attributes. In the line of text below, an empty pair of span tags with the ID “hourstoday” is where the script will insert a line of text.

Today's Hours: <span id="hourstoday"></span>

For the div tag below, the ID “announcement” ties both CSS styling and JavaScript control to this division.

<div id="announcement">Special announcement message</div>

Near the bottom of the page, a script tag links the web page to scripts in a separate document. The script has to be placed below any code in the page that it has to act on.

<script src="js/libraryhours.js"></script>

**JavaScript Coding**

This is contents of the libraryhours.js file. It begins with comments of things to remember when I update the file. The first note is especially important.

// Months are 0-11. Date is actual date.
//
// For break periods, write "closed until" date in the isItBreak message and
// bottom of page message as the first day the library is again open.

Next is the definition of the function called “isItHoliday”. The “var today = new Date();” line tells the browser to get the current date from the computer and make it something called “today”. The codes today.getMonth() and today.getDate() extract the month and date from “today”. It then compares the
current date with a list of holiday dates that you set up at the beginning of the semester. In JavaScript, the double-equal sign is the equal operator, the double-ampersand is AND, the double pipe is OR, and parentheses are used for nesting. The function gives a true or false answer to: Is today’s month September AND is the date the 5th, OR is it November AND the 8th, OR is it November AND the 11th, etc.

```javascript
function isItHoliday() {
  var today = new Date();
  return (today.getMonth() == 8 && today.getDate() == 5) // Labor Day
  || (today.getMonth() == 10 && today.getDate() == 8) // General Election Day
  || (today.getMonth() == 10 && today.getDate() == 11) // Veterans' Day
  || (today.getMonth() == 10 && today.getDate() == 24) // Thanksgiving
  || (today.getMonth() == 10 && today.getDate() == 25); // Day after Thanksgiving
}
```

The function called “isItBreak” checks whether the current date is within a break period like Spring Break or the break between academic sessions. It introduces two operators: >= for greater than or equal to, and <= for less than or equal to. Here it would have answered true if the current date was both on or after August 13th AND on or before August 21st.

```javascript
// Allows special message for a range of days. Start with Saturday before break
// If break spans two months, specify a range for each month and OR them
function isItBreak() {
  var today = new Date();
  return (today.getMonth() == 7 && today.getDate() >= 13) &&
          (today.getMonth() == 7 && today.getDate() <= 21);
}
```

In a similar manner, these functions check to see if the current date is during one of our end-of-semester extended hours days.

```javascript
function isItSpecial1() {
  var today = new Date();
  return (today.getMonth() == 3 && today.getDate() == 27) ||
          (today.getMonth() == 3 && today.getDate() == 28);
}

function isItSpecial2() {
  var today = new Date();
  return (today.getMonth() == 4 && today.getDate() == 2) ||
          (today.getMonth() == 4 && today.getDate() == 3) ||
          (today.getMonth() == 4 && today.getDate() == 4) ||
          (today.getMonth() == 4 && today.getDate() == 5);
}
```

We now run these four functions we just defined in a series of If...Else statements. If isItHoliday is true, then our span tag in the web page with the ID “hourstoday” will have the phrase “Holiday - Closed” inserted into it. If isItHoliday is not true, then we move on to isItBreak, and then the others if necessary.

```javascript
if (isItHoliday())
    {document.getElementById("hourstoday").innerHTML = "Holiday - Closed";}
else if (isItBreak())
    {document.getElementById("hourstoday").innerHTML = "Closed until Aug 22";}
else if (isItSpecial1())
    {document.getElementById("hourstoday").innerHTML = "Special Hours: 7:30 am - 9:00 pm";}
else if (isItSpecial2())
    {document.getElementById("hourstoday").innerHTML = "Special – 7:30 am - 8:00 pm";}
else {
    // Allows special message for a range of days. Start with Saturday before break
    // If break spans two months, specify a range for each month and OR them
    var today = new Date();
    return (today.getMonth() == 7 && today.getDate() >= 13) &&
            (today.getMonth() == 7 && today.getDate() <= 21);
}
```

If all four functions answer false, then our Else turns to a switch statement. This switch statement asks for the current day of the week (which comes back as a number from 0 to 6, with 0 being Sunday), and
defines “dailyhours” as a phrase depending on what day of the week it is. The last line inserts the dailyhours phrase into the hourstoday span in the web page.

```javascript
var dailyhours;
switch (new Date().getDay()) {
    case 0:
        dailyhours = "Sun &ndash; Closed";
        break;
    case 1:
        dailyhours = "Mon &ndash; 7:30 am - 7:00 pm";
        break;
    case 2:
        dailyhours = "Tue &ndash; 7:30 am - 7:00 pm";
        break;
    case 3:
        dailyhours = "Wed &ndash; 7:30 am - 7:00 pm";
        break;
    case 4:
        dailyhours = "Thurs &ndash; 7:30 am - 7:00 pm";
        break;
    case 5:
        dailyhours = "Fri &ndash; 7:30 am - 3:30 pm";
        break;
    default:
        dailyhours = "Sat &ndash; Closed";
        break;
}
document.getElementById("hourstoday").innerHTML = dailyhours;
}
```

The dailyhours statements include the day of the week, followed by an n-dash. This is because mobile devices often don’t automatically refresh web pages. Including the day can alert the viewer that they are looking at information that is several days old.

The script below reuses our isItBreak function to control whether the library hours table at the bottom of the page shows the regular hours, or shows a special message during a break period. The big difference between this script and the one above is that instead of just inserting a line of text, it is inserting HTML tags and text. This is done so that the hours can be shown in two table columns and the special message as a line that spans both columns. Be aware that because the HTML tags include double quotes, we must use single quotes around the code string you wish to insert, instead of the customary double quotes.

```javascript
// This controls the Regular Hours table at the bottom of the page
if(isItBreak())
{document.getElementById("hoursbriefrow1").innerHTML = '<td colspan="2">Closed until Fall Semester</td>';document.getElementById("hoursbriefrow2").innerHTML = '<td colspan="2">starts on Monday Aug 22.</td>';
} else {
document.getElementById("hoursbriefrow1").innerHTML = '<td class="hoursltcol">Mon-Thur</td><td class="hoursrtcol">7:30 am - 7:00 pm</td>';
document.getElementById("hoursbriefrow2").innerHTML = '<td class="hoursltcol">Fri</td><td class="hoursrtcol">7:30 am - 3:30 pm</td>';
}
```

The script below uses the same general technique to control when a special announcement box is displayed on the library home page. Instead of inserting a line of text onto a visible part of the page, it uses a slightly different command control an attribute within an HTML tag.

The message is entered into a division with id “announcement” on the web page in advance of it being needed. The division has its display attribute specified as “display:none”, hiding it from view. The script
overrides this setting, changing it to display:block during the active period. This way, you can manually hide or reveal the message by editing the attribute, or have it appear based on the date.

// This makes the special announcement appear within a certain date range

function showAnnouncement() {
    var today = new Date();
    return (today.getMonth() == 10 && today.getDate() >= 1) &&
    (today.getMonth() == 10 && today.getDate() <= 11);
} if(showAnnouncement()) {
    document.getElementById("announcement").style.display = "block";
}

Bonus Code Sample

We also have code that changes the background image on the library web pages based on the time of day. It works by changing the class of the division that contains that background. The image files are specified in the CSS.

function imageByHour() {
    var hour = new Date().getHours();
    if (hour < 6) {
        document.getElementById("pageback").setAttribute("class", "background3");
    } else if (hour < 11) {
        document.getElementById("pageback").setAttribute("class", "background1");
    } else if (hour < 18) {
        document.getElementById("pageback").setAttribute("class", "background2");
    } else {
        document.getElementById("pageback").setAttribute("class", "background3");
    }
}

imageByHour()