IST400/600 Scripting for Games, Spring 2010
Lab 5
Instructor: Keisuke Inoue

Description:

In this lab, you will create the Bus Schedule Web page that was discussed in the previous class. The following instructions will help you to create the page, but will not tell you all the small steps. The lab is designed to encourage your learning and enhance your understanding by working independently. **You may ask questions to instructors and/or friends (but keep it quiet) and look up previous lecture materials, textbook, and other resources. Do not copy-and-paste any resources other than you created.**

Peer Survey:

We don't do the peer survey this time. (But we will, next week!) But you will still need to submit your work to MySite, and post the URL to the discussion board.

Description:

The Bus Schedule displays the times of next buses for two bus stops (for inbound/outbound, eastbound/westbound, etc), based on the current time.

Implementation Outline:

1. Interface: Displaying the name of the two bus stops and the times of next buses.
2. Logic: Looking through arrays to find the next bust time, and convert the time to an appropriate format.

Lab Instructions:

1. Interface

Displaying the bus stops, destination, and time.
Create a new file on your text editor, and save it with a name "Lab5.html" (or whatever you like). As always, insert <script> and <body> elements.

Specify "updateBustTime();" for the onLoad attribute of the body. This function will be defined later, to update and display the bus times.

Now, select at least two combinations of bus stops and destinations that you want to display from the following:

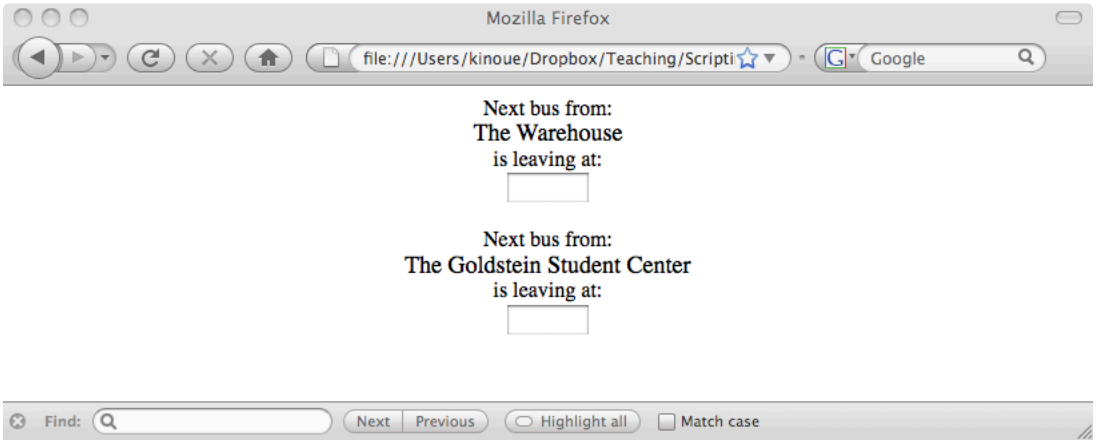
Bus Stops:
- Goldstein Student Center
- Manley South
- College Place
- SUNY-ESF
- Syracuse Stage
- Center of Excellence
- The Warehouse

Destinations:
- Goldstein Student Center
- The Warehouse

You can say something like “Next bus to <the destination of your choice> is leaving from <the bus stop of your choice> is leaving at:”. If you are using a bus stop that is one of the destinations, you don’t have to say the destinations, since it is obvious.

Below each text, insert a text field, with name: “busTime1” and “bustTime2” (or something like that). If you would like, use the size attribute, to set the width of the text field (“6” worked good with me.)

At this point, open the file with FireFox. It should look like below:



2. Logic

2.1) Defining arrays

The arrays have been already defined in separate files stored under http://kinoue.mysite.syr.edu/IST400-600. You can just use the <script> with the src attribute to include the files in the top of your page. The files are named systematically and the name of the global variables defined are the same as the file name (with out “.js”:

gBusTimes00.js	To The Warehouse from Goldstein
gBusTimes01.js	To The Warehouse from Manley South
gBusTimes02.js	To The Warehouse from College Place
gBusTimes03.js	To The Warehouse from SUNY-ESF
gBusTimes04.js	To The Warehouse from Syracuse Stage
gBusTimes05.js	To The Warehouse from Center of Excellence
gBusTimes11.js	To Goldstein from Manley South
gBusTimes12.js	To Goldstein from College Place
gBusTimes13.js	To Goldstein from SUNY-ESF
gBusTimes14.js	To Goldstein from Syracuse Stage
gBusTimes15.js	To Goldstein from Center of Excellence
gBusTimes16.js	To Goldstein from The Warehouse

For example, if you are looking for a bus schedule from College Place to Goldstein, you will include gBusTimes12.js by inserting the following line at the top of your page:

```
<script src=http://kinoue.mysite.syr.edu/IST400-600/gBusTimes12.js></script>
```

This will define a global array variable `gBusTimes12` for you.

2.2) Define updateBusTime() Function

Now, define the updateBusTime() function. The purpose of this function is to find the next bus times, given the current time. Our strategy is to look through the arrays by using for-loops, and compare each bus-leaving time with the current time. Once we find a time that is later than now, we use that time to display it. In the definition of the function, do the following:

1. Declare a variable “dateTime”, and initialize the variable by calling “new Date();”.

2. Declare a variable “mins”, and initialize it by calling “dateTime.getMinutes()”;

3. Declare a variable “hours”, and initialize it by calling “dateTime.getHours()”;

4. Declare a variable “timeNow”, and initialize it by using “mins” and “hours”. This variable should contain the minutes passed in the day. (Remember the formula?)

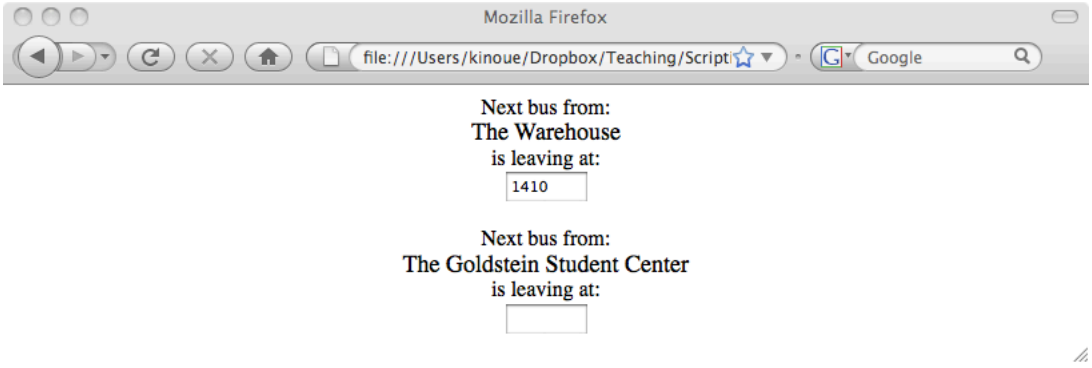
5. Declare a variable “timeForBus”, and initialize it with “Not found yet” (or whatever you like).

6. Now, create a for-loop statement that goes through one of the arrays.

7. In the for-loop, compare each element in the array with the timeNow variable. If timeNow is “earlier” than the time in the array, update the timeForBus variable with the element.

8. Just for testing, update the “bustTime1” text field with the timeForBus variable.

At this point, let’s see if it’s working so far. Reload FireFox and see if it working… you should be able to see something like this:



This good, but you have one small problem that happened during the class, too. The problem is that the for-loop goes until the end, and kept updating the timeForBus variable, so the variable always ends up with the last bus time in the array. Before moving on, fix the problem. If it is showing the real bus time, it should be showing sometime between about 1050 (5:30pm) and 1140 (7:00pm). There are a few ways to fix it. (Hint: You can add another condition to the if statement, or you the “break” statement.)

2.3) Define formatTime() Function

Now, let’s convert the timeForBus to a format which is easier to understand. In doing so, let’s define a function that does the job (since you will need to do it again later anyway.) Define a function “formatTime()”, below the updateBusTime() function. Insert one parameter, “myTime” (or whatever) for the function.

In the function, do the following:

1. Declare a variable “myHours”, and initialize it by dividing myTime by 60 and “flooring” the outcome by using “Math.floor()”.

2. Declare a variable “myMins”, and initialize it by calling “time % 60”. (Remember, ‘%’ is called the “modulo” operator, which returns the remainder when the preceding number is divided by the later number.)

3. Declare a variable “myAmPm”, and initialize it with “’AM’”

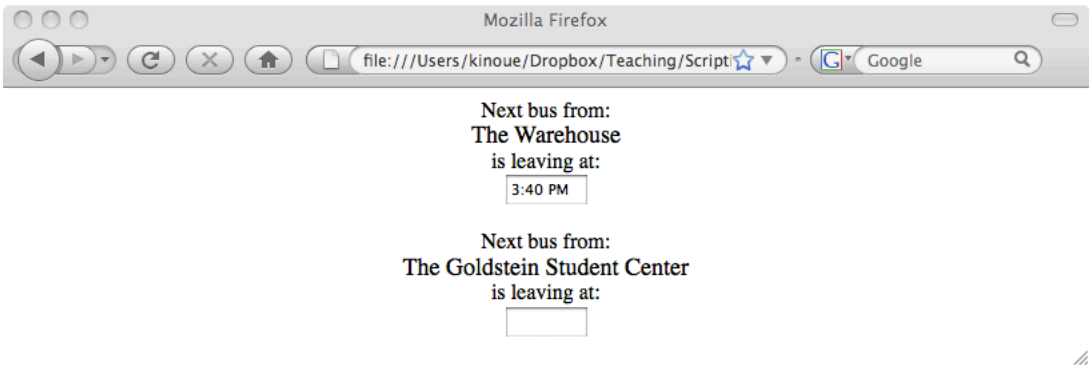
4. Modify myAmPm and myHours, so that they will show appropriate time all together. Specifically:

    a. If myHours is more than 12, myHours should be subtracted by 12, and myAmPM should be “PM”.
    b. If myHours is 12, myAmPm should be “PM”.
    c. Other wise, no modification is needed.

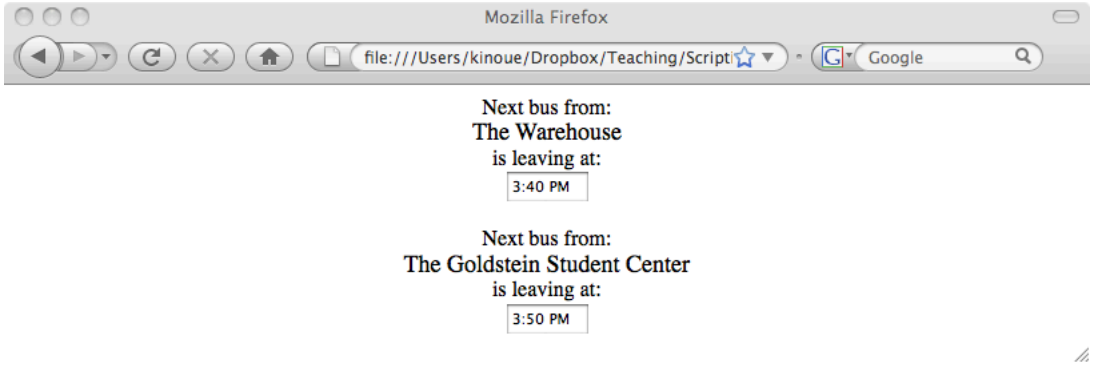
5. Return the concatenation of the string. In other words, insert the following statement in the last line of the function definition.

```
return myHours + “:” + myMins + “ “ + myAmPm;
```

Now, you can use “formatTime(myBusTime)” instead of just myBusTime to update the text field. Reload FireFox, and make sure it is working.



If you have gone this far, just repeat the step 5) to 8) in the 2.2) Define updateBusTime() Function, and you are done! Save the file and reload FireFox. Make sure both of the bus times are displayed correctly.
(If you would like, you can make sure by looking up:
http://connectivecorridor.syr.edu/wp-content/uploads/2007/01/CC-centro-schedule_Jan2010_Final_web.pdf)



Congratulations! If you are undergraduate, **move on the “Submission” Section in the next page**. If you are a graduate student, **move on to the “Extra Task for IST Graduate Students” Section.**

Extra Task for IST Graduate Students

As an extra task, IST graduate students need to modify the so that the user can select the bus stop of his/her interests, by following the instructions below:

1. In the top of the <body> element, insert a <form> element (with whatever the name you like), and insert a drop down menu, by using the <select> element in the form.

   Make sure to the onChange attribute to "updateBusTime();". Also make sure that each value represent the array index for each bus stopThe selected attribute of the first option element defines the default choice. It should look something like below. (You can copy-and-paste it, but I strongly encourage you to type them, for your learning.)

   ```
   <select name="busSelector" onchange="updateBusTime();" >
       <option selected="1" value="0">Goldstein Student Center</option>
       <option value="1">Manley South</option>
       <option value="2">College Place</option>
       <option value="3">SUNY-ESF</option>
       <option value="4">Syracuse Stage</option>
       <option value="5">Center of Excellence</option>
       <option value="6">The Warehouse</option>
     </select>
   ```

   Modify the text on the page accordingly.

2. In the top of the page, import the two two-dimensional arrays, instead of the two single-dimension arrays used previously, by changing the file names to "gBusTimes0.js" and "gBusTimes1.js". Each of the files contains all the times for all the stops to one destination. (You probably guessed it… gBusTimes0 contains gBusTimes00, gBusTimes01, gBusTimes02,… and so on.)

   Remember, gBusTimes0.js defines a two dimensional array called gBusTimes0 and gBusTimes1.js defines a two dimensional array called gBusTimes1.

3. Modify the updateBusTime() function accordingly. It is actually very simple! Hint: Use the value property of the <select> element as an index, for choosing a single-dimensional array from the two-dimensional array. (Ok, it sounds complicated, but it is not that bad, actually… seriously.)

Submission

As always, submit your work to MySite. The Instruction of MySite is found at http://its.syr.edu/mysite/.

You will only need to submit the URL to page you created to the discussion board on the ILMS. Create a hyperlink, if you would like. For example, I would submit:

http://kinoue.mysite.syr.edu/IST400-600/BusSchedule.html

Grading Criteria

The lab assignment must be submitted by Tuesday Mar. 2nd midnight. The grading will be based on the following criteria:

- Presentation: 2 points
- Logic: 2 points
- For-loop: 2 points
- Converting the time 2 points
- HTML syntax: 2 points

Late submission will be accepted within one week from the original due date, with 2 points deduction. 2 points will be deducted for improper submissions (e.g. not using MySite) or for not submitting the peer survey as well.