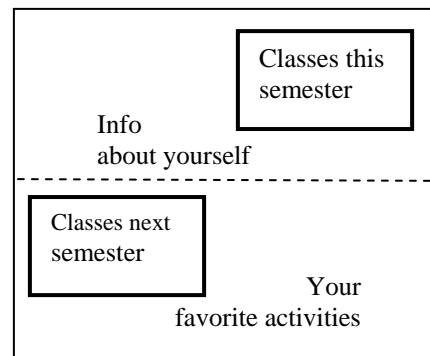


COMP 322 Internet Systems Fall 2007 Assignment 2

Due in the digital drop box Monday, Sept. 17 by 11:00 PM.

1. Create an HTML document that presents the initial part of the class syllabus, down through the **Objectives** heading. The rendering need not be formatted like my version; you can quite easily get something that looks better. But this is not the point. The point is to get experience with CSS techniques. Use an internal stylesheet (using a `style` element in the document head), but also link to an external stylesheet (a file with extension `.css`). The external stylesheet should import (use the `@import` command) another style sheet (another file with extension `.css`). Define rules for at least five elements that appear in the body of the HTML document. And define at least two classes that are used in the body. Include at least three rules in the internal stylesheet, at least three rules in the external style sheet to which the HTML document links, and at least two rules in the imported stylesheet. Include at least one example where a property value is overridden and at least one example where it isn't.

2. Use the techniques of section in the Course Notes entitled "Text Flow and Box Model" to format a document as shown in the figure at right. There are four elements shown here. Two of them float (one right, one left) and have visible borders. (Use some of the available properties to get interesting effects.) Two of them flow, one around and down from the first box, the second around and down from the second box. The dotted line in the middle of the figure should not appear in the rendering; it is simply the conceptual line between the first floating element and the second floated element.



How text flows around floated elements will depend on how you have your browser window dimensioned. Try to make the flowing elements big enough and the floating, boxed elements small enough so that, with a browser screen of reasonable size, we see flowing text pass under the bottom of the floated text.

3. Write an HTML document that uses JavaScript to input the average miles per gallon of four cars on three test runs and displays these figures, along with the best car on each run and the best run for each car, as in the table shown below. Use prompt dialogue boxes for input. You don't have to input names for the cars or the test runs; just identify them with "Car 1" or "Test run 1" and so on. Do input validation to ensure that values are numbers and are greater than zero. Use a **do ... while** loop. Note that, if you apply `parseFloat` to a string that cannot be interpreted as number, it returns the special value `NaN`, which is detected with the function `isNaN`. For example,

```
isNaN( parseFloat( "A1" ) )
```

returns `true`. More simply, apply `isNaN` directly to the string. For example, `isNaN("A1")` returns `true` while `isNaN("12")` returns `false`. Use separate functions for input, finding the best run for each car, finding the best car for each run, and outputting the table. (You may define other functions.) Be sure to use loops where they are natural. Write code that works even if the number of cars or the number of test runs changes. Note that the function that outputs the table must have a loop (for the columns) inside a loop (for the rows).

	Test run 1	Test run 2	Test run 3	Test run 4	Best
Car 1	35.6	32.2	39.2	41.3	Run 4
Car 2	42.8	39.8	45.2	41.2	Run 3
Car 3	21.2	15	25	22.4	Run 3
Car 4	30.6	26.2	36.2	28.8	Run 3
Best	Car 2	Car 2	Car2	Car 1	

4. Write an HTML document that uses JavaScript to input two strings. (Use the `prompt` method of the window object to raise prompt dialog boxes.) It then outputs the words that occur in both strings. Case is not significant in determining whether words are different. (For example, “in” and “In” count as the same word.) A word is not output more than once even if it occurs several times in one or both of the strings. We assume that words are delimited by spaces. For example, suppose that the following two strings are input:

In the cold room
The cat in the room

Then the following words should be output

in the room

To do this, convert both the input strings to all lowercase and split them at the spaces, giving two arrays of lowercase words. Remove duplicate entries. Then use a nested loop structure to compare all words in the first array against all words in the second array; output the elements that are equal. You can use the following code to remove duplicate elements in an array. You can also download this code from the same place you got this assignment.

```
function removeDups( ar )
{
    var i = 0;

    while ( i < ar.length ) {
        var j = i + 1;

        while ( j < ar.length && ar[i] != ar[j] )
            j ++;

        if ( j < ar.length ) {
            for ( var k = i; k < ar.length-1; k ++ )
                ar[k] = ar[k+1];

            ar.length--;
        }
        else
            i ++;
    }
}
```