

Announcements

- ❖ Check class announcements daily
- ❖ You must implement programming projects by yourself

Form Processing Review

- ❖ **Example:** FormExample folder
 - ❖ Notice the use of <fieldset> and <legend>
 - ❖ Notice the <label>

Assignment #5

- ❖ Let's talk about some parts of this project

Scrolling Tables

- ❖ You can separate the table sections by using
 - ❖ `thead`, `tfoot`, `tbody`
- ❖ For large tables, when printed, each page will show the header and the footer
- ❖ Elements must be defined in the order: `thead`, `tfoot`, `tbody`
- ❖ **Example: ScrollingTables.html**

Sorting

- ❖ Note → strings can be compared using < >. For example:

```
var x = "a";  
var y = "b";  
var z = x < y;
```

- ❖ sort() function
 - ❖ sort data using alphabetical order
- ❖ **Example:** Sorting.html

Math Constants

- ❖ `Math.E` → Euler's constant
- ❖ `Math.PI`
- ❖ `Math.SQRT1_2` → Square root of $\frac{1}{2}$
- ❖ `Math.SQRT2` → Square root of 2
- ❖ `Math.LN2` → Natural log of 2 (~0.693)
- ❖ `Math.LN10` → Natural log of 10 (~2.302)
- ❖ `Math.LOG2E` → Base 2 log of E (~1.442)
- ❖ `Math.LOG10E` → Base 10 log of E (~0.434)
- ❖ **Example:** `MathConstants.html`

Math Functions

- ❖ Math.min → returns the smallest of a series of numbers
- ❖ Math.max → returns the maximum of a series of numbers
- ❖ Math.round → rounds to nearest integer ($\geq .5$ to next integer, $< .5$ to previous integer)
- ❖ Complete listing at:

https://developer.mozilla.org/en/Core_JavaScript_1.5_Reference/Global_Objects/Math

Array Functions

- ❖ Modify the original array
 - ❖ push → adds element(s) to array and returns length of new array
 - ❖ reverse → reverse order of elements
 - ❖ splice → adds/or removes elements
 - ❖ shift → removes the first element and returns it
 - ❖ unshift → adds element(s) to the front of array
- ❖ Do not modify the original array
 - ❖ indexOf → returns index of first element in the array corresponding to the argument
 - ❖ lastIndexOf → returns the last index of an element in the array corresponding to the argument
 - ❖ join → joins all elements into a string
 - ❖ slice → extracts a section of array
 - ❖ concat → returns array with elements from current array and another array
- ❖ Complete listing at:

https://developer.mozilla.org/en/Core_JavaScript_1.5_Reference/Global_Objects/Array

String Object

- ❖ Created using new String()
- ❖ Wraps the primitive string data type and add methods
- ❖ Methods
 - ❖ charCodeAt → returns Unicode value for the character
 - ❖ indexOf → returns index of first occurrence of specified argument value
 - ❖ lastIndexOf → returns index of last occurrence of specified argument value
 - ❖ concat → combines two strings
 - ❖ quote → wraps string in quotes
 - ❖ split → splits the string into an array of strings
 - ❖ substr → returns a substring starting at an index position and extending a number of characters
 - ❖ substring → returns a string between two indexes
 - ❖ toLowerCase
 - ❖ toUpperCase
- ❖ Complete listing at:

https://developer.mozilla.org/en/Core_JavaScript_1.5_Reference/Global_Objects/String

Recursion

- ❖ A recursive function is one that “calls itself”.
- ❖ One approach to see this process is to visualize that, instead of the function calling itself, the function calls another function that performs the same task as the original one.
- ❖ Nature has several examples of recursive phenomena.
- ❖ A typical recursive example is the computation of the factorial.
- ❖ Definition of factorial (non-recursive)

$$n! = n * n-1 * n-2 * \dots 1$$

Definition of factorial (recursive)

$$n! = n * (n-1)!$$

- ❖ Let's implement a recursive factorial function.
- ❖ Computation of the fibonacci series is another example of recursion
- ❖ Let's draw a call tree that illustrates the set of calls that are taking place.