Tutorial 10 Notes

Arrays (slides 10 to 16)

* An array is a collection of values organized under a single name
* The index is the number that each individual value is associated with and that distinguishes it from other values in the array
* Array values are referenced using the expression array[i] where array is the name of the array and i is the index of a specific value in the array, index values start with zero

Array declaration (slide 17)

* var array = new Array(length);
* array is the name, length is optional
* Array’s current size property is array.length

Setting a value for an array item (or the entire array)

* array[i] = value;
* var monthName = new Array(“January”, “February”, “March”, “April”, “May”, “June”, “July”, “August”, “September”, “October”, “November”, “December”);
* var x = [“April”, 3.14, true, null];

Changing the order of the array items (slides 19 and 20)

* reverse() reverses the order of items in an array, making the last items first and the first items last
* sort() rearranges array items in alphabetical order
* The sort method when applied to numeric values will sort the values in order by their leading digits, rather than by their numerical values

Extracting and inserting array items (slides 25 and 26)

* To create a subarray use slice() method - array.slice(start, stop)where start is the index value of the array item at which the slicing starts and stop is the index value at which the slicing ends
* The stop value is optional; if it is omitted, the array is sliced to its end
* splice() method removes and inserts array items - array.splice(start, size, values)where start is the starting index in the array, size is the number of array items to remove after the start index, and values is an optional comma-separated list of values to insert into the array

Using arrays as data stacks (slides 27 to 30)

* array.push()
* array.pop()
* array.shift()
* array.unshift()

Program loops (for, while, do) (slides 31 to 36)

for (start value, condition, increment/decrement) {

statements

}

for (var i=1; i<=5; i++) {

statements

}

while (condition) {

statements

}

do {

statements

} while (condition);

Comparison and logical operators (slides 37 and 38)

* Comparison operators: ==, ===, !=, >, >=, <, <=
* Logical operators: && (and), || (or), ! (not)

Accessing each value in an array (and built-in array methods)

for (var i=0; i < array.length; i++) {commands involving array[i]}

* JavaScript supports several methods to loop through the contents of an array without having to create a program loop structure
* Each of these methods is based on calling a function that will be applied to each item in the array
* The general syntax is array.method(callback [, thisArg]) where array is the array, method is the array method, and callback is the name of the function that will be applied to each array item
* forEach() method runs a function for each item in the array (slide 43)

Other array methods

* map(), filter(), every(), some() (summarized in slide 47)

Conditional statements (if, if else, else if) (slides 49 to 53)

* if (condition) {statements}

Modulus operator %

break and continue (slides 61 and 62)

* break statement terminates any program loop or conditional statement
* continue statement stops processing the commands in the current iteration of the loop and continues on to the next iteration

**Lyman Hall Theater August Schedule** (lht\_august.html, lht\_events.js and lht\_calendar.js programs)