

ANNOUNCEMENTS

- ❖ No posting of code in the forum
- ❖ Check class announcements daily

REVIEWING ONE-DIMENSIONAL ARRAYS

- ❖ How do we define an array?
- ❖ How do we represent an array?
- ❖ How can we access the elements of an array?
- ❖ What can we do with the elements of an array?
 - ❖ Everything we can do with a variable
- ❖ Which iteration statement is frequently used with arrays?
- ❖ Fundamental loop you should remember

```
for (idx = 0; idx < a.length; idx++) { /* task */ }
```

where `a` is an array
- ❖ Arrays are created in a memory area called the heap
- ❖ Array variable holds address of array
- ❖ How are array elements accessed?
- ❖ We can create aliases to arrays via assignments
- ❖ Arrays are objects
 - ❖ Object → Entity that has values and operations (functions)

NULL

- ❖ What is null?
 - ❖ Represents no value
 - ❖ Represents no address
 - ❖ `var a = null;`
- ❖ **Example:** Null.html
- ❖ When can use null?

NULL AND UNDEFINED

- ❖ null → indicates no value
- ❖ undefined
 - ❖ Value associated with uninitialized variables
 - ❖ `var x; //` in a function
 - ❖ When a function that is expected to return a value does not returns one (IMPORTANT case)
 - ❖ Value associated with object properties that do not exist
- ❖ `==` considers null and undefined equal
- ❖ `===` considers null and undefined different

NAN

- ❖ **NaN** → Not-A-Number (Same as Number.NaN)
 - ❖ Unequal to any number including itself
 - ❖ Use isNaN function → determines (returns true or false) whether an argument is not a number. It attempts to convert the argument to a number
 - ❖ The following comparisons return false
NaN == NaN, NaN === NaN
- ❖ To remember → **!**isNaN() allow us to determine whether an expression is a number
 - ❖ Notice: isNaN(20) → False
 - ❖ You may want to write a function call isNumber that returns !isNaN(x)
- ❖ **Example:** NaN.html

VALIDITY CHECK

- ❖ **Example:** ValidityCheck.html
 - ❖ Notice that using Number was not necessary; Why?

IN-CLASS LAB (20 MINUTES)
