Review (the last session covered repetition statements)

* Arrays: Multiple storage elements, elements are accessed by an index
* Loops: A for loop repeats a block of code some set number of times. It has a built in counter

Intro

* Project 2 instructions are now on Canvas. Project 2 uses functions
* No class on Monday

Functions

* Functions are reusable modules that let us use the same code over and over without re-writing it.
* Functions used in programming are similar to functions in math. They have input values (called *parameters*), processing is done on the input values, and an output (*return value*) is produced.   
  Example:
  + Math function: *f(x) = 2x*
  + JavaScript function: function double(x) { return x \* 2; }
* *Declaring* (defining) vs. *invoking* (calling) a function:
  + Function definition: function add(x, y) { return x + y; }
  + Function call: var sum = add(3, 2);
* Functions you’ve already used. Examples of calling them:
  + var num = parseInt(someInput);
  + var result = Math.sqrt(9);
* Functions don’t always have a return value:
  + namesArray.push(“Rey”);
  + alert(“You win!”);
* Functions don’t always have parameters:
  + var name = namesArray.pop();

Scope

Scope is the area of the program in which a variable is defined.

* Global scope: the variable is available for use everywhere  
  Example:
  + var a = 5;  
    var b = 2;  
    var result = 0;  
    function add(x, y) {  
     result = x + y;  
     return result;  
    }  
    alert(add(a, b));  
    alert(result); // result is defined
* Local scope: the variable is available only in the function where it was declared  
  Example:
  + var a = 5;  
    var b = 2;  
    function add(x, y) {  
     var result = x + y;  
     return result;  
    }  
    alert(add(a, b));  
    alert(result); // result is undefined
* Best practice: avoid using global variables inside of functions. Use arguments and return values.
  + Analogy: Using global variables inside a function is like trying to run a store with multiple entrances and exits. Sales people might not see when customers come in, and worse yet, customers could leave without going through the checkout line and without paying.
  + Example of bad practice- using global variables with a parameterless function:  
    var a = 5;  
    var b = 2;  
    var result;  
    function add() {  
     result = a + b;  
    }  
    alert(result); // this works, but it’s bad practice