

## Introduction to Arrays

- ❖ Grouping of similarly named variables, which are grouped sequentially in memory and accessed by their element (*index*) number
- ❖ Element numbering begins with 0 to one less than the total number of elements
- ❖ An Array element can hold numbers, strings, Boolean (true/false), and Objects

Counter[0]	30
Counter[1]	45
Counter[2]	53
Counter[3]	2
Counter[4]	879

Copyright © 2005 R.M. Laurie 1

## Declaring Arrays

- ❖ Declaration:
 

Counter[0]	30
Counter[1]	45
Counter[2]	53
Counter[3]	2
Counter[4]	879

  - ◆ `var Counter = new Array(5);`
    - ◆ Reserves Counter array memory Counter[0] to Counter[4]
    - ◆ No values are stored in elements
    - ◆ May store assign values to elements individually
 

```
Counter[0] = 30;
Counter[1] = 45;
...
```
  - ◆ `var Counter = new Array(30, 45, 53, 2, 879);`
    - ◆ Reserves Counter array memory Counter[0] to Counter[4] and initialized the first 5 elements to the the values shown

Copyright © 2005 R.M. Laurie 2

## for Loop Array Initialization

- ❖ A for loop can be used to initialize a declared array
- ❖ Set all array elements to 0
 

```
var Counter = new Array(5);
for(var K=0; K< 5 ; K++)
    Counter[K] = 0;
```

Counter[0]	0
Counter[1]	0
Counter[2]	0
Counter[3]	0
Counter[4]	0
- ❖ This is very useful for large arrays such as:
 

```
var Score= new Array(100);
for(var K=0; K< 100 ; K++)
    Score[K] = 0;
```

Copyright © 2005 R.M. Laurie 3

## Array Bounds Checking

- ❖ For JavaScript the array element quantity is optional. The following is acceptable syntax.
 

```
var Counter = new Array();
```
- ❖ Elements can be added to an existing Array by assigning values to new array elements. The number of elements is increased to eight.
 

```
var Counter = new Array(5);
for(var K = 0; K < 8; K++)
    Counter[K] = 0;
```
- ❖ The array length property specifies the total number of elements contained in an array.
 

```
for(var K=0; K< Counter.length; K++)
    Counter[K] = 0;
```

Copyright © 2005 R.M. Laurie 4

