

Indexing: Just More Of The Same

**CSE
100**

Giving a unique name to things complicates referring to them. Indexing simplifies reference by allowing a single base name to be “specialized” by a number.

© Larry Snyder, 1999

**CSE
100**

Indexing, The Basic Idea

- ❖ Motivation: When there is a large number of similar things that must be referenced and manipulated, it can be inconvenient to think up a unique name for each, and to refer to them by the name

+ For example: Each day in May could have been given a different name, but these would be hard to remember and inconvenient to use

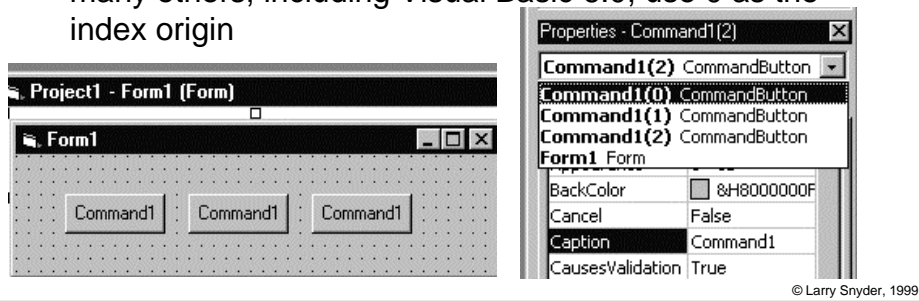
- ❖ Indexing names the items by associating a base name and a number -- the index -- with each one
- ❖ Computer notation: May(5)

Streets and Avenues
Superbowls
Congresses

© Larry Snyder, 1999

CSE 100 Indexing Particulars

- ❖ In everyday indexing, it is common to begin the indexing with 1, e.g. May 1, SuperBowl I, etc.
- ❖ The number at which indexing begins is known as the *origin*
- ❖ Many computer languages use 1 as the origin, but many others, including Visual Basic 6.0, use 0 as the index origin



CSE 100 Arrays

- ❖ A collection of like things named using indexing is called an *array*
- ❖ Arrays are used for representing collections of data values, e.g. integers, strings, etc.

For example:

```
colorList(0) = "red"  
colorList(1) = "white"  
colorList(2) = "blue"
```

- ❖ The elements of an array must all be of the same type
- ❖ The index of an array element is also known as a *subscript*

Arrays In VB6.0

- ❖ Arrays are declared like any other variable using a Dim statement

Keyword Array name Largest Index Type
 → → → →
 Dim colorList(2) As String

```
colorList(0)
colorList(1)
colorList(2)
```

- ❖ Notice

- ✦ The syntax is just like a normal declaration except for the parenthesis pair
- ✦ In the parentheses is the largest desired index
- ✦ The total number of elements of the array will be one more than the largest index, since the origin is 0
- ✦ The type applies to all of the elements

Indexing Arrays

- ❖ To refer to different elements of the array, it is necessary only to change the index ...

```
Form1.Line (100,100)-(500,200),QBColorByName(colorList(0)),B
Form1.Line (100,200)-(500,300),QBColorByName(colorList(1)),B
Form1.Line (100,300)-(500,400),QBColorByName(colorList(2)),B
```

```
colorList(0) = "red"
colorList(1) = "white"
colorList(2) = "blue"
```



- ❖ The index value must be an integer constant (1), a variable (**myNdex**) or expression (**myNdex+1**)

**CSE
100**

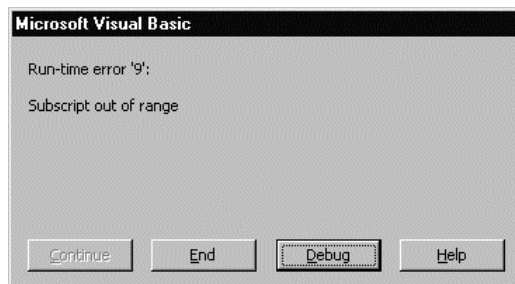
Combining Indexing, Arrays, Loops

- ❖ To sweep through the elements of an array, use a loop

```
For i = 0 To 2
    Form1.Line (100,i*100+200)-(500,i*100+300),
                QBColorByName(colorList(i)),B
Next i
```



- ❖ A common error is to is to index beyond the end of the array ...



© Larry Snyder, 1999