Using Arrays and Collections Transcript

Ever need to build a collection of related information? Perhaps you need your user to declare the locations where they plan to use a product. Maybe you need to manage a list of names. An array is a powerful data structure that enables this.

At the end of this course, you'll be able to explain:

- what arrays are and why they are useful.
- Recognize commonly used array snap blocks and how to use them.
- Select from the various loop functions to work with arrays.
- And identify how the system uses arrays in Multi-Select fields.

Objectives

- Definition
- Operations
- Loops
- Multi-Select

Definitions

An Array, also called a Collection, is a data structure that contains zero, one, or many elements. Each of these elements are identified by an index.

Each item in an array must be the same data type. Think number, text string, or Boolean true/false.

This means that you can have an array of numbers, or an array of text strings. You cannot have an array that stores both.

The index of arrays is zero based. Meaning it starts at zero and counts up from there. An element positioned at n has an index of n-1… Consider the element in the third spot. Three minus one is two, so it has an index of two.

Computer programs often leverage arrays to organize data and keep related information grouped. They structure that data so it is easily sorted, searched, and modified.

You can perform loop functions on an entire array. This executes that process on each member of that array according to the function's parameters.

Think of a queue line full of customers as an array. While there are customers in that line, the clerk processes their transactions in order.

The clerk continues performing that action as long as there are customers in the line… even if new customers keep getting added.

That line is the array, the customers are the data, and the clerk represents the system performing a while loop on the array.

While there is at least one customer in line, the clerk processes the next transaction.

Select the right arrow icon for a brief knowledge check.

Operations

Snap has a variety of blocks that enable you to get data from and alter the data in an array. Follow along with the example of a text array for customersInLine.

Begin by declaring your Array in your workspace.

Add the declare block to your code, name it, and choose the type of array you want. In this case we're creating customersInLine as a Text Array.

Note the symbol after text. It appears as a small rectangle but is actually an open bracket followed by a closed bracket symbol. These symbols indicate an array, and their data is often represented in between the brackets separated by commas.

Leave the set to empty for now. We call an array like this without any data an empty array.

With your array declared, you can use the Add item To Array block to add items to it.

With this block, add the string block with the name Moe… to the array called… customersInLine.

The Get block is the most common way to get a handle to the entire array.

That adds one element to the array. It can be repeated for our other two customers: Larry and Shemp.

Now that I have my array created and populated, I can perform operations on or with it.

As before, the Get block will help us get a handle to the entire array.

We could get an item from a specific index of the array. Like, Get Item at index 1 from customersInLine.

The get first or last item in block looks at whichever end of the array you select and gives you that item.

The Length of array block counts the number of entries in the array and returns that as a number. This example would return the number 3.

You can use the search block to examine the contents of an array. It supports 3 kinds of searches. Any, Filter, and Find.

Any returns a Boolean true if any items meet the search criteria, and returns false if not.

Filter returns a new subset array of all items in the original array that match the search criteria.

Find returns first item in the array that matches the search criteria.

Use these variations of the search block to quickly examine the content of your arrays for specific data.

Leverage the Empty array block to remove all data from it. The array will still exist, but it won't contain any elements.

There are many other array operators you can use to control array. Explore the array section of code blocks and review the documentation for all available options.

Select the right arrow icon to continue to the Loops topic.

Loops

Often to work with an array, you'll use a loop to interact with each item in it.

The most common is the For each loop.

Add the loop to your snap code, then select the array for it to work though.

The for each loop will loop through each element in the array and make a temporary variable, called Item, to make that element easy to process.

For example, we can scan the names of all the customers in line to find the customer with the shortest name.

Now when your code runs, it will perform that code on each item in your array.

You can use two other loops to work through an array as well, but they're a little more complicated.

You can count through a loop, using the variable I to refer to the position of an element.

This might be used if you only want to perform the code on a limited number of items in your array.

The while loop is more flexible. It will continue to process the code within it as long as a conditional statement is true. Be careful with this loop. If the condition you set for the loop will never be false, it creates an infinite loop: a task that will never complete. When this happens, the user's browser produces an error saying the page is unresponsive.

Use these three loops to manage and manipulate your arrays. You'll find that by far the most useful loop is the for each loop.

Select the right arrow Icon to move to the next topic.

Multi-Select

Now that you understand what an array is, you may start to recognize arrays throughout the Epicor CPQ platform.

For example, the pages in your configurator are stored as an array of pages.

The fields in your user interface are stored as an array.

You can even store an array of text, numbers, or other data in a field.

To use a field that has a control type of Multi-Select, you must first set the Data Type to an array.

Then, the data in this field creates an array of that data type.

You can then use this array in other parts of your configurator. Here, we'll create Validation Rule that makes sure the user selected at least one option in the multi-select.

Declare a variable as a number, then set that variable to the length of array.

Use the Get Field to pull the array from the UseLocation field.

Now we can use that count of selections in a Validate Field to make sure that the countOfSelections is greater than zero.

If it's not, we'll return the message "Please Select at least one option." To the user.

Select the right arrow icon to continue to a final knowledge check.

Recap

An array is a powerful data structure that enables you to sort and group data.

You should now be able to explain what an array is and why they are useful.

Recognize commonly used array snap blocks and how to use them.

Select from the various loop functions to work with arrays.

And identify how the system uses arrays in Multi-Select fields.

About Epicor

Epicor Software Corporation drives business growth. We provide flexible, industry-specific software that is designed around the needs of our manufacturing, distribution, retail, and service industry customers. More than 40 years of experience with our customers' unique business processes and operational requirements is built into every solution—in the cloud, hosted, or on premises. With a deep understanding of your industry, Epicor solutions spur growth while managing complexity. The result is powerful solutions that free your resources so you can grow your business. For more information, connect with Epicor or visit www.epicor.com.

EPICOR

Corporate Office
804 Las Cimas Parkway
Austin, TX 78746
USA
Toll Free: +1.888.448.2636
Direct: +1.512.328.2300
Fax: +1.512.278.5590

Latin America and Caribbean
Blvd. Antonio L. Rodriguez #1882 Int. 104
Plaza Central, Col. Santa Maria
Monterrey, Nuevo Leon, CP 64650
Mexico
Phone: +52.81.1551.7100
Fax: +52.81.1551.7117

Europe, Middle East and Africa
No. 1 The Arena
Downshire Way
Bracknell, Berkshire RG12 1PU
United Kingdom
Phone: +44.1344.468468
Fax: +44.1344.468010

Asia
238A Thomson Road #23-06
Novena Square Tower A
Singapore 307684
Singapore
Phone: +65.6333.8121
Fax: +65.6333.8131

Australia and New Zealand
Suite 2 Level 8,
100 Pacific Highway
North Sydney, NSW 2060
Australia
Phone: +61.2.9927.6200
Fax: +61.2.9927.6298