Introduction to LINQ

Imagine you're building a new game in C#, with dozens of characters to manage in your database. How would you access them all? What if you need to apply a filter? What if you needed to format each character name?

You might think of storing characters in a list and running through each element with a `foreach` loop. You'd have to write nested `if` statements, re-format each element, and store each result in a new list.

The result isn't pretty.

Suppose that we want to find all the names in a list which are longer than 6 letters and return them in all uppercase letters. You can see what it would look like in **Program.cs** in the code editor.

And remember that this only works in a running C# file. What if the database was stored in a separate server somewhere and it was implemented with SQL instead of C#?

The solution is *LINQ*. It works for complex selections and transformations, and it works on local and remote data sources. Each selection/transformation is called a *query*, and LINQ gives us new syntax and methods to write them.

Imagine LINQ like an add-on to C# and .NET. Once you import the LINQ features, you can write new syntax, like:

```
string[] names = { "Tiana", "Dwayne", "Helena" };
var filteredNames = from n in names
  where n.Contains("a")
  select n;
```

And you can use new methods on collections, like `Where()`:

```
var shortNames = names.Where(n => n.Length < 4);
```

In this lesson you'll learn :

How to import the LINQ features to C#

How to run LINQ queries on datasets

How to identify method and query syntax

Basic operators, such as `Select`, `Where`, and `from`

☑Instructions

1.
In **Program.cs**, compare the two approaches for querying data:

Without LINQ, we use a `foreach` loop and nested `if` statement.

With LINQ, we write a three-line query.

Run the code to see them in action!