**LINQ (Language-Integrated Query)**:

- A feature in C# that allows querying collections (arrays, lists, databases, XML, etc.) in a readable and concise way using a SQL-like syntax.   
- LINQ provides a consistent way to query data from different sources like databases (LINQ to SQL), XML (LINQ to XML), and collections (LINQ to Objects).

A computer code with many colorful text

Description automatically generated with medium confidence A black background with white text

Description automatically generated

- Need to import using line at top of our code:



- Often times we use LINQ with generic collections (like lists), so you may see both namespaces imported into a file:

A screen shot of a computer

Description automatically generated

- Every LINQ query returns either a single value or an object of type IEnumerable<T>.   
- For now, all you need to know about that second type is that:

* It works with foreach loops, just like arrays and lists.
* You can check its length with Count().

- Since the single value type and/or the parameter type T is not always known, it’s common to store a query’s returned value in a variable of type var.  
- var is just an implicitly typed variable — we let the C# compiler determine the actual type for us. Here’s one example:

A computer screen shot of a code

Description automatically generated

**Method and Query Syntax:**

- **Query syntax** looks like a multi-line sentence similar to SQL:  
- With query syntax, the query keywords must be all lowercase.

A screenshot of a computer

Description automatically generated

- **Method Syntax** looks like regular C# code:  
-For method syntax, the [methods](https://www.codecademy.com/resources/docs/c-sharp/methods) begin with a capital letter.

A screenshot of a computer program

Description automatically generated

**Basic Query Syntax:**

- A basic LINQ query, in query syntax, has three parts:   
- The from and select [operators](https://www.codecademy.com/resources/docs/c-sharp/operators) are required, where is optional.

* The **from operator** declares a variable to iterate through the sequence. In this case, h is used to iterate through heroes.
* The **where operator** picks elements from the sequence if they satisfy the given condition. The condition is normally written like the conditional expressions you would find in an if statement. In this case, the condition is h.Length < 7.
* The **select operator** determines what is returned for each element in the sequence. In this case, it’s just the element itself.

A computer screen shot of text

Description automatically generated A black background with text and symbols

Description automatically generated with medium confidence

**Basic Method Syntax: Where:**

- In method syntax, each query operator is written as a regular method call.  
- In the last exercise, we selected every element with a length under 7. Here it is in method syntax:

A computer code with text

Description automatically generated

- The where operator is written as the method Where(), which takes a lambda expression as an argument.   
- Lambda expressions are a quick way to write a method. In this case, the method returns true if h is less than 7 characters long.  
- Where() calls this lambda expression for every element in heroes. If it returns true, then the element is added to the resulting collection.

**Basic Method Syntax: Select:**

- To transform each element in a sequence — like writing them in uppercase — we can use the select operator.  
- In method syntax it’s written as the method Select(), which takes a lambda expression:

A screenshot of a computer code

Description automatically generated

- We can combine Select() with Where() in two ways:

1. Use separate statements:

A screenshot of a computer program

Description automatically generated

1. Chain the expressions:

A screenshot of a computer

Description automatically generated

- You can imagine each line like a step in a conveyor belt, filtering and transforming the sequence as it goes.

**When To Use Each Syntax:**

1. For single-operator queries, use the method syntax.
2. For everything else, use the query syntax.