

Metodo	Categoria	Descrizione	Esempio
Where	Filtraggio	Filtrà gli elementi di una sequenza in base a una condizione booleana	<pre>var adults = people.Where(p => p.Age >= 18);</pre>
Select	Proiezione	Trasforma ogni elemento della sequenza in un nuovo formato	<pre>var names = people.Select(p => p.Name);</pre>
SelectMany	Proiezione/Flatten	Appiattisce sequenze di sequenze in un'unica sequenza	<pre>var allOrders = customers.SelectMany(c => c.Orders);</pre>
OrderBy	Ordinamento	Ordina la sequenza in ordine crescente secondo una chiave	<pre>var sorted = people.OrderBy(p => p.Name);</pre>
OrderByDescending	Ordinamento	Ordina la sequenza in ordine decrescente secondo una chiave	<pre>var sortedDesc = people.OrderByDescending(p => p.Age);</pre>

ThenBy / ThenByDescending	Ordinamento multiplo	Ordina secondariamente dopo un OrderBy	<pre>people.OrderBy(p => p.LastName).ThenBy(p => p.FirstName);</pre>
GroupBy	Raggruppamento	Raggruppa gli elementi secondo una chiave	<pre>var groups = people.GroupBy(p => p.City);</pre>
Join	Join	Unisce due sequenze sulla base di una chiave comune	<pre>var result = customers.Join(orders, c => c.Id, o => o.CustomerId, (c,o) => new {c.Name, o.OrderId});</pre>
GroupJoin	Join	Join con raggruppamento dei dati correlati	<pre>var result = customers.GroupJoin(orders, c => c.Id, o => o.CustomerId, (c,os) => new {c.Name, Orders = os});</pre>
Any	Test / Condizione	Restituisce true se almeno un elemento soddisfa la condizione	<pre>bool hasAdults = people.Any(p => p.Age >= 18);</pre>

All	Test / Condizione	Restituisce true se tutti gli elementi soddisfano la condizione	<code>bool allAdults = people.All(p => p.Age >= 18);</code>
Contains	Test / Condizione	Controlla se la sequenza contiene un elemento specifico	<code>bool hasJohn = names.Contains("John");</code>
Count	Aggregazione	Conta il numero di elementi, optionalmente filtrati	<code>int adultsCount = people.Count(p => p.Age >= 18);</code>
Sum	Aggregazione	Somma i valori di una sequenza	<code>int totalAge = people.Sum(p => p.Age);</code>
Min / Max	Aggregazione	Restituisce il valore minimo o massimo di una sequenza	<code>int youngest = people.Min(p => p.Age);</code>
Average	Aggregazione	Calcola la media dei valori di una sequenza	<code>double avgAge = people.Average(p => p.Age);</code>
First / FirstOrDefault	Elemento singolo	Restituisce il primo elemento che soddisfa la condizione (o default)	<code>var firstAdult = people.FirstOrDefault(p => p.Age >= 18);</code>

Last / LastOrDefault	Elemento singolo	Restituisce l'ultimo elemento che soddisfa la condizione (o default)	<pre>var lastAdult = people.LastOrDefault(p => p.Age >= 18);</pre>
Single / SingleOrDefault	Elemento singolo	Restituisce l'unico elemento che soddisfa la condizione (o default)	<pre>var unique = people.SingleOrDefault(p => p.Id == 1);</pre>
Take / TakeWhile	Selezione	Prende i primi N elementi o finché la condizione è vera	<pre>var top3 = people.Take(3);</pre>
Skip / SkipWhile	Selezione	Salta i primi N elementi o finché la condizione è vera	<pre>var afterFirst3 = people.Skip(3);</pre>
Distinct	Set	Restituisce solo elementi unici	<pre>var uniqueCities = people.Select(p => p.City).Distinct();</pre>
Union	Set	Unisce due sequenze eliminando duplicati	<pre>var allNames = names1.Union(names2);</pre>

Intersect	Set	Restituisce elementi presenti in entrambe le sequenze	<code>var commonNames = names1.Intersect(names2);</code>
Except	Set	Restituisce elementi presenti nella prima sequenza ma non nella seconda	<code>var onlyInFirst = names1.Except(names2);</code>
Concat	Set	Unisce due sequenze senza eliminare duplicati	<code>var combined = names1.Concat(names2);</code>
Reverse	Ordinamento	Inverte l'ordine della sequenza	<code>var reversed = people.Reverse();</code>
ToList / ToArray	Materializzazione	Converte la sequenza in lista o array, eseguendo la query	<code>var list = people.ToList();</code>
Aggregate	Aggregazione	Esegue una funzione cumulativa su tutta la sequenza	<code>int product = numbers.Aggregate((a,b) => a*b);</code>
DefaultIfEmpty	Sequenza	Restituisce una sequenza con un valore	<code>var result = emptyList.DefaultIfEmpty(0);</code>

di default se vuota

OfType

Filtro tipo

Filtrare elementi di un certo tipo

```
var strings =  
objects.OfType<string>();
```