

A LINQ (Language Integrated Query) operátorok széles skáláját kínálja, amelyek különböző műveletek végrehajtására szolgálnak az adatgyűjteményeken. Az alábbiakban felsorolom a leggyakrabban használt LINQ operátorokat:

### 1. Szűrési operátorok (Filtering)

- **Where:** Szűri az elemeket egy adott feltétel alapján.

csharp

Kód másolása

```
var result = collection.Where(x => x > 10);
```

- **Find:** Kiszűri egy elemet egy adott feltétel alapján.

csharp

Kód másolása

```
var result = collection.Find(x => x > 10);
```

- **OfType:** Kiválasztja az adott típusú elemeket.

csharp

Kód másolása

```
var result = collection.OfType<int>();
```

### 2. Átalakítási operátorok (Projection)

- **Select:** Átalakítja az elemeket egy másik típusra.

csharp

Kód másolása

```
var result = collection.Select(x => x.Name);
```

- **SelectMany:** Egy összetett adatstruktúrából (pl. lista a lista belső elemeivel) laposítja le a gyűjteményt.

csharp

Kód másolása

```
var result = collection.SelectMany(x => x.Items);
```

### 3. Csoportosító operátorok (Grouping)

- **GroupBy:** Csoportosítja az elemeket egy adott kulcs alapján.

csharp

Kód másolása

```
var result = collection.GroupBy(x => x.Category);
```

#### 4. Rendezési operátorok (Ordering)

- **OrderBy:** Növekvő sorrendbe rendezi az elemeket.

csharp

Kód másolása

```
var result = collection.OrderBy(x => x.Name);
```

- **OrderByDescending:** Csökkenő sorrendbe rendezi az elemeket.

csharp

Kód másolása

```
var result = collection.OrderByDescending(x => x.Name);
```

- **ThenBy:** Másodlagos rendezés, növekvő sorrend.

csharp

Kód másolása

```
var result = collection.OrderBy(x => x.Name).ThenBy(x => x.Age);
```

- **ThenByDescending:** Másodlagos rendezés, csökkenő sorrend.

csharp

Kód másolása

```
var result = collection.OrderBy(x => x.Name).ThenByDescending(x => x.Age);
```

#### 5. Aggregáló operátorok (Aggregation)

- **Sum:** Az elemek összegét számolja.

csharp

Kód másolása

```
var result = collection.Sum(x => x.Amount);
```

- **Average:** Az elemek átlagát számolja.

csharp

Kód másolása

```
var result = collection.Average(x => x.Score);
```

- **Min:** Az elemek legkisebb értékét találja meg.

csharp

Kód másolása

```
var result = collection.Min(x => x.Price);
```

- **Max:** Az elemek legnagyobb értékét találja meg.

csharp

Kód másolása

```
var result = collection.Max(x => x.Price);
```

- **Count:** Megszámolja az elemeket.

csharp

Kód másolása

```
var result = collection.Count(x => x.IsActive);
```

- **Distinct:** Eltávolítja az ismétlődő elemeket.

csharp

Kód másolása

```
var result = collection.Distinct();
```

## 6. Összehasonlító operátorok (Comparison)

- **Any:** Ellenőrzi, hogy van-e olyan elem, amely megfelel egy feltételnek.

csharp

Kód másolása

```
var result = collection.Any(x => x > 10);
```

- **All:** Ellenőrzi, hogy az összes elem megfelel-e egy feltételnek.

csharp

Kód másolása

```
var result = collection.All(x => x > 0);
```

- **Contains:** Ellenőrzi, hogy egy adott elem benne van-e a gyűjteményben.

csharp

Kód másolása

```
var result = collection.Contains("apple");
```

- **SequenceEqual:** Ellenőrzi, hogy két gyűjtemény egyenlő-e (mindkét gyűjtemény elemei és sorrendje is megegyezik).

csharp

Kód másolása

```
var result = collection.SequenceEqual(otherCollection);
```

## 7. Kombináló operátorok (Set Operations)

- **Union:** Két gyűjtemény uniója (duplikációk nélkül).

csharp

Kód másolása

```
var result = collection1.Union(collection2);
```

- **Intersect:** Két gyűjtemény metszete.

csharp

Kód másolása

```
var result = collection1.Intersect(collection2);
```

- **Except:** Két gyűjtemény különbsége (az első gyűjtemény elemei, amelyek nem szerepelnek a második gyűjteményben).

csharp

Kód másolása

```
var result = collection1.Except(collection2);
```

## 8. Egyéb operátorok

- **First:** Az első elem visszaadása, amely megfelel a feltételnek (vagy hiba, ha nem található).

csharp

Kód másolása

```
var result = collection.First(x => x > 10);
```

- **FirstOrDefault:** Az első elem visszaadása, ha van, egyébként alapértelmezett érték.

csharp

Kód másolása

```
var result = collection.FirstOrDefault(x => x > 10);
```

- **Last:** Az utolsó elem visszaadása, amely megfelel a feltételnek (vagy hiba, ha nem található).

csharp

Kód másolása

```
var result = collection.Last(x => x > 10);
```

- **LastOrDefault:** Az utolsó elem visszaadása, ha van, egyébként alapértelmezett érték.

csharp

Kód másolása

```
var result = collection.LastOrDefault(x => x > 10);
```

- **Single:** Egyetlen elem visszaadása, amely megfelel a feltételnek (vagy hiba, ha több találat van).

csharp

Kód másolása

```
var result = collection.Single(x => x.Id == 5);
```

- **SingleOrDefault:** Egyetlen elem visszaadása, ha van, egyébként alapértelmezett érték (vagy hiba, ha több találat van).

csharp

Kód másolása

```
var result = collection.SingleOrDefault(x => x.Id == 5);
```

## 9. Végrehajtási operátorok (Execution)

- **ToList:** A gyűjteményt listává alakítja.

csharp

Kód másolása

```
var result = collection.ToList();
```

- **ToArray:** A gyűjteményt tömbbé alakítja.

csharp

Kód másolása

```
var result = collection.ToArray();
```

- **ToDictionary:** A gyűjteményt szótárrá alakítja.

csharp

Kód másolása

```
var result = collection.ToDictionary(x => x.Id);
```

- **AsEnumerable:** Visszaadja az adatokat IEnumerable típusúként (általában nem szükséges, de segíthet a típusok kezelésében).

csharp

Kód másolása

```
var result = collection.AsEnumerable();
```

Ezek az operátorok segítenek abban, hogy az adatokat különböző módokon szűrjük, átalakítsuk, rendezzük, összesítsük és manipuláljuk C#-ban, miközben deklaratív módon dolgozunk velük.