

# LINQ

Mon, Mar 16, 2020 [Linq](#)

## LINQ

### Tijdelijke notities:

De onderstaande LINQ artikelen zijn allemaal reeds gedaan:

- Alle AND, OR, BETWEEN, Less..., Greater..., Equals.
- Alle NnnnConditionExpressions, Alle NnnnConstantExpressions, Alle nnnnNvlExpressions. (Dus Blob, boolean, integer, long, string)
- Alle Convert en alle CanConvert
- 

LINQ (Language-Integrated Query) is an important innovation in .NET (since version 3.5). There is a gap between the imperative programming style and the functional programming style. LINQ helps bridge that gap. In other words: LINQ bridges the gap between the world of objects and the world of data.

See the list of all more than 130 [LINQ expressions](#).

### What is the difference?

An **imperative** language uses a sequence of statements to determine how to reach a certain goal. These statements are said to change the state of the program as each one is executed in turn.

The **functional** programming paradigm was explicitly created to support a pure functional approach to problem solving. Functional programming is a form of declarative programming.

### When is functional useful?

Functional languages are good when you have a fixed set of things and as your code

evolves, you primarily add new operations on existing things. This can be accomplished by adding new functions which compute with existing data types and the existing functions are left alone.

The AI-Framework is rooted in the functional approach. Therefore LINQ is a very useful means of accessing data. LINQ lightens the programmer's job, because it allows the programmer to focus on the **business logic**.

LINQ has a wide range of potential applications that provide a declarative model for retrieving and processing data. This allows the programmer to use the same syntax with a wide range of various types of data. In other words: LINQ is independent of a data source like databases, XML documents, various Web services, arrays, and so on.

### Why LINQ is great to use

- The LINQ query expression (unlike a traditional SQL statement) is strongly typed. Therefore, the C# compiler ensures that these expressions are syntactically well formed.
- Query expressions have metadata representations within the assembly that makes use of them. Tools such as the recommended Visual Studio 2010 IDE can use this metadata for useful features such as IntelliSense, auto completion and so forth.
- Because LINQ queries are fully name and type-checked at compile-time, runtime error surprises are minimal.

### Quick access

- [AndExpression](#)
- [BetweenExpression](#)

- [BlobConditionExpression](#)
- [BlobConstantExpression](#)
- [BlobNvlExpression](#)
- [BooleanConditionExpression](#)
- [BooleanConstantExpression](#)
- [BooleanNvlExpression](#)
- [CanConvertToDateTimeExpression](#)
- [CanConvertToDecimalExpression](#)
- [CanConvertToEnumExpression](#)
- [CanConvertToIntegerExpression](#)
- [CanConvertToLongExpression](#)
- [ConvertToBlobExpression](#)
- [ConvertToDateTimeExpression](#)
- [ConvertToDecimalExpression](#)
- [ConvertToEnumExpression](#)
- [ConvertToIntegerExpression](#)
- [ConvertToLongExpression](#)
- [ConvertToQrCodeExpression](#)
- [DateTimeNvlExpression](#)
- [DateTimeAdditionExpression](#)
- [DateTimeConditionExpression](#)
- [DateTimeConstantExpression](#)
- [DateTimeConstructExpression](#)
- [DateTimeMaxExpression](#)
- [DateTimeMinExpression](#)
- [DateTimeSubtractDaysExpression](#)
- [DateTimeSubtractExpression](#)
- [DayOfTheMonthExpression](#)
- [DayOfWeekExpression](#)
- [DaysBetweenExpression](#)
- [DecimalAbsExpression](#)
- [DecimalAdditionExpression](#)
- [DecimalCeilingExpression](#)
- [DecimalConditionExpression](#)
- [DecimalConstantExpression](#)
- [DecimalFloorExpression](#)
- [DecimalMaxExpression](#)

- [DecimalMinExpression](#)
- [DecimalMultiplyExpression](#)
- [DecimalNvlExpression](#)
- [DecimalQuotientExpression](#)
- [DecimalRoundExpression](#)
- [DecimalSubtractionExpression](#)
- [EnumConditionExpression](#)
- [EnumConstantExpression](#)
- [EnumNvlExpression](#)
- [EqualsExpression](#)
- [FirstDateOfWeekExpression](#)
- [GreaterThanExpression](#)
- [GreaterThanOrEqualExpression](#)
- [HourExpression](#)
- [InListExpression](#)
- [IntegerAbsExpression](#)
- [IntegerAdditionExpression](#)
- [IntegerBitwiseAndExpression](#)
- [IntegerConditionExpression](#)
- [IntegerConstantExpression](#)
- [IntegerMaxExpression](#)
- [IntegerMinExpression](#)
- [IntegerModuloExpression](#)
- [IntegerMultiplyExpression](#)
- [IntegerNvlExpression](#)
- [IntegerQuotientExpression](#)
- [IntegerSubtractionExpression](#)
- [IntegerToCharExpression](#)
- [IsNullExpression](#)
- [IsValidEmailAddressExpression](#)
- [IsValidHostNameExpression](#)
- [LinqAggregateDateTimeExpression](#)
- [LinqAggregateDecimalExpression](#)
- [LinqAggregateIntegerExpression](#)
- [LinqAggregateLongExpression](#)
- [LinqConcatExpression](#)
- [LinqDistinctExpression](#)

- [LinqElementAtExpression](#)
- [LinqExistsExpression](#)
- [LinqOrderByExpression](#)
- [LinqSearchExpression](#)
- [LinqSelectExpression](#)
- [LinqSkipExpression](#)
- [LinqTakeExpression](#)
- [LinqThanByExpression](#)
- [LinqUnionExpression](#)
- [LinqWhereByTypeExpression](#)
- [LinqWhereExpression](#)
- [LongAdditionExpression](#)
- [LongConditionExpression](#)
- [LongConstantExpression](#)
- [LongMaxExpression](#)
- [LongMinExpression](#)
- [LongModuloExpression](#)
- [LongMultiplyExpression](#)
- [LongNvlExpression](#)
- [LongQuotientExpression](#)
- [LongSubtractionExpression](#)
- [MinuteExpression](#)
- [MonthExpression](#)
- [NotExpression](#)
- [NowExpression](#)
- [OrExpression](#)
- [ReasonIsNullOrEmptyExpression](#)
- [RegexMatchesExpression](#)
- [SecondsExpression](#)
- [StringConcatExpression](#)
- [StringConditionExpression](#)
- [StringConstantExpression](#)
- [StringContainsExpression](#)
- [StringConversionExpression](#)
- [StringGetHashCodeExpression](#)
- [StringGetLengthExpression](#)
- [StringIndexOfExpression](#)

- [StringIsNullOrEmptyExpression](#)
- [StringJoinExpression](#)
- [StringLastIndexOfExpression](#)
- [StringLeftExpression](#)
- [StringMatchesExpression](#)
- [StringNvlExpression](#)
- [StringPadLeftExpression](#)
- [StringPadRightExpression](#)
- [StringReplaceExpression](#)
- [StringRightExpression](#)
- [StringStartOrEndsWithExpression](#)
- [StringSubstringExpression](#)
- [StringTruncateExpression](#)
- [TodayExpression](#)
- [TotalTicksUntilExpression](#)
- [TotalTimeUntilExpression](#)
- [WeekExpression](#)
- [YearAndWeekExpression](#)
- [YearExpression](#)

Online URL: <https://wiki-ai-framework.abstract-it.nl/article/linq-39.html>