DTI Guide: C3 Expressions

C3 includes an expression evaluation system to make it easy to build simple functions. Essentially, they are one-line Java expressions.

C3 Expressions are used all over the C3 AI Suite.

- Filter fields
- Projection expressions
- Data Transforms
- Dataset Processing
- Timeseries Metric expressions
- etc...

The C3 AI Suite supports a java-like expression syntax allowing the user to define complex functions quickly and easily throughout the Suite. This pseudo-language supports basic arithmetic and boolean operators as well a large set of built-in functions known as the ExpressionEngineFunctions.

Here are some useful techniques and syntax which is available in C3 Expressions, but which may not be easy to find in current documentation

Supported Syntax

- **Comparisons**
  - Expressions like a > b, c >= d which return boolean values
- **Ternary Operators** which return a value based on the result of a boolean expression
  - <conditional_expression> ? <value_if_true> : <value_if_false>
- **Java basic math operators**
  - Addition (+)
  - Subtraction (-)
  - Multiplication (*)
  - Division (/)
  - Modulus (%)
  - etc...
- **Java Standard Libraries and Functions**
  - Ex: Math - Math.abs, Math.cos, etc...
  - Type casting - ex. dateTime('2020-01-01')
- **ExpressionEngineFunctions (C3 defined functions)**
  - rolling(aggFunc, input_series, ...) - Computes a rolling aggregation of the input timeseries
  - identity(value) - Produces a new timeseries consisting of repeating entries of 'value'.
  - eval(aggFunc, interval, input_series, ...) - Forces evaluation of the input timeseries at the set interval, and aggregation with aggFunc.
  - Most ExpressionEngineFunctions are designed to work with timeseries data

Special keywords

`this`

Sometimes, it may be useful for a timeseries to refer to itself, or to get a reference to the 'current' timeseries. This is done with the this name. For example, we can specify the `transform` field of a `TsDecl` metric to do some transformation of the data before it heads to normalization. We can use the `fillMissing` function to fill gaps in the data with a specified value. We'd specify this with `fillMissing(this, <value_to_fill>)`.

Esspecially Useful Expressions

Ternary Operators

Ternary Operators are widely useful throughout the C3 AI Suite. They allow small conditional expressions which can affect the return value of your expression.

Timeseries expressions

- rolling
  - computes a rolling aggregation over a timeseries. It takes an aggregation function, a timeseries (which will be aggregated), and possibly another timeseries to signal when to restart the aggregation. rolling is like an expanding window function which may be dynamically reset.
- identity
  - takes a value, and simply repeats this value when building the timeseries. it's useful occasionally if you need a timeseries consisting entirely of one value.
- eval
  - generates a timeseries using a specific start/end date. This is useful for generating timeseries that rely on window functions which may rely on values before the start date of a requested timeseries. Essentially, eval builds an entire timeseries which is passed to the next function or on to the next step in the normalization process.
- rollingdiff

    - returns a time series in which every value is computed by taking the difference between current and previous point.
- fillMissing
    - Will impute missing values with some default value you specify. This is useful if you want to indicate missing values in some special way.

Official Documentation

- General topic page: https://developer.c3.ai/docs/7.19.0/topic/metrics-expression-engine-functions-home
- A list of ExpressionEngineFunctions is available
    - Online: https://developer.c3.ai/docs/7.19.0/type/ExpressionEngineFunction
    - Through the Static Console: c3ShowType(ExpressionEngineFunction)