

# Introduction to BRAIN Expression Language

---

## Table of Contents

1. What is Fast Expression?
2. Characteristics of Fast Expression
  - Data fields
  - Operators
3. Further Knowledge of Fast Expression

## What is Fast Expression?

“Fast expression” is a proprietary programming language used by WorldQuant BRAIN that is designed to make it easier to write and test financial models. The language can be thought as a form of pseudo code, which uses natural language and simple programming constructs to convey the logic of the algorithm.

The goal of using “Fast expression” on BRAIN is to provide a clear and concise way to express complex ideas and algorithms that can be easily understood by other developers and researchers. By abstracting away the details of the underlying implementation, it can allow BRAIN users to focus on the high-level logic of their algorithms, rather than getting bogged down in the implementation details.

## Characteristics of Fast Expression

Just like how an English sentence consists of a subject, verb and object; Fast expression can include data fields, operators and numerical values.

## Data Fields

Data fields refer to a named collection of data, for example 'open price' or 'close price'.

Dataset	Field
Company Fundamental Data for Equity	1,575
US News Data	644
...	...
Relationship Data	82
Options Analytics	74
Price Volume Data for Equity	72

A dataset containing price, volume, close, open price for each stock

Field	Description	Type
adv20	Average daily volume in past 20 days	Matrix
close	Daily close price	Matrix
returns	Daily returns	Matrix
volume	Daily volume	Matrix

## Operators

Operators refer to a set of mathematical techniques required to implement your Alpha ideas.

### 1. Arithmetic Operators

### 2. Vector Operators

### 3. Logical Operators

### 4. Transformational Operators

### 5. Cross Sectional Operators

### 6. Group Operators

### 7. Time Series Operators

Arithmetic Operator	Description
+, -, *, / , ^	Arithmetic operators: add, subtract, multiply, divide, power
log(x)	Natural logarithm

Cross Series Operator	Description
rank(x, n)	Ranks the input among all the instruments and returns an equally distributed number between 0.0 and 1.0

Time Series Operator	Description
ts_rank(x, n)	Rank the values of x for each instrument over the past n days, then return the rank of the current value. The rank value is between 0.0 and 1.0

```

1  /*
2  HYPOTHESIS: If a stock closes below its open price more often in the last month compared to the last year, there could be a reversal in the stock
   price and it may increase in the short term.
3
4  IMPLEMENTATION: Buy more of stocks for which such days occur more often (ts_sum) in last month (20 days), compared to the last year (250 days).
5
6  HINT: Can introducing the intensity of daily stock fluctuations or price instability improve the alpha?
7  */
8
9  a = ts_sum(open>close,20)/ts_sum(open<close,250);
10
11 b = ts_sum(open>close,250)/ts_sum(open<close,250);
12
13 rank(a/b)
14

```

- `/*` helps to create block comments that span multiple lines of text, while `*/` denotes the end of the comment. Comments consist of explanatory text to help understand what the code does. [1]
- `;` (semicolon) acts as a semicolon in a sentence, separating the end of one sentence from the beginning of another sentence. For the last line of the code (line 13) `;` is not needed. [2]
- The last sentence of the entire expression is the alpha expression that the BRAIN simulator use to calculate the positions to take in each stock. [3]

Lastly, Fast expression does not have classes, objects, pointers, or functions.

In summary, Fast expression provides a clear and concise way for users to express complex ideas and algorithms. Don't worry if you're not familiar with Fast expression yet. With a bit of practice, we believe you'll pick it up in no time!