**Standard Fx**

**Fx is a language which design to use in some small occasion. What grammar it supports is easy for anyone.**

**You can get start very quickly, just continue reading.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sign** | | **Form** | **Description** | **Note** |
| **#** | | **Consist of + or - or \* or / or ^ or < or > or =** |  |  |
| **Ei** | **V** | **Consist of Aa-Zz and \_ but no beginning with \_** | **An variable** |  |
| **C** | **? V=E0** | **A function** | **? V=E1 in E0 is illegal**  **V in E0 must refers the value what it will received**  **As soon as it received a value, it return E0's value** |
| **Consist of 0-9 and at most one . and e or e- in it** | **A number** |  |
| **Consist of chars in ""** | **A string** | **"" means "** |
| **\_true** | **A bool** | **If E0's value is \_true then what statement expressed by E0 is true** |
| **\_false** | **If E0's value is \_false then what statement expressed by E0 is false** |
| **{}** | **A list** | **An empty list** |
| **{E1,E2,...,En}** | **A list that has n(n>=0) members** |
| **D** | **(E1 E2)** | **Call function** | **E1 must be a function and received E2 and return a value as (E1 E2)'s value**  **If left char is ( or { or , or : then write E1 E2 instead of E1 E2** |
| **(E1^E2)** | **E1 to the power of E2** |  |
| **(E1\*E2)** | **Multiply E1 by E2** |  |
| **(E1/E2)** | **E1 divided by E2** |  |
| **(E1+E2)** | **E1 plus E2** |  |
| **(E1-E2)** | **E1 subtract E2** |  |
| **(E1<E2)** | **E1 less than E2** |  |
| **(E1>E2)** | **E1 greater than E2** |  |
| **(E1=E2)** | **E1 equals to E2** |  |
| **(E1/=E2)** | **E1 not equals to E2** |  |
| **(E1<=E2)** | **E1 less than or equals to E2** |  |
| **(E1>=E2)** | **E1 greater than or equals to E2** |  |