Standard Fx

Fx is a language which design to use in some small occasion:

For example, you can print "Hello World" like this:

```
print("Hello World")
```

A function you write could insert in any right places and means the same function in the same application's source code, like this function:

```
?(x){x < 0:f(x),x = 0:g(x),x > 0:t(x)}
```

You can create an object like this:

```
pair{fst:0,snd:1}
```

Here is the definition of Fx:

Sign		Form	Description	Note	
#		Consist of + or – or * or / or ^ or < or = or > or & or |	An operator	There is no priority in operators	
N		Consist of any chars	A note	() in N must be paired	
T_i		Consist of Aa-Zz or _	A type	If T_i beginning with _ then T_i is the standard type else T_i is the user type	
E_i	V_i	Consist of Aa-Zz or _ but no beginning with _	An identifier	V_i is an variable in default The variable V_i refers the global variable V_i's value in default	
	C	$?(V_1,V_2,...,V_m):\{E_{-1}:E_1, E_{-2}:E_2,..., E_{-n}:E_n\}$	A function	Note	Type
				The variable V_0 in E_{-j} or E_j must refers the global variable V_0's value(It is Law A) The variable V_i in E_{-j} or E_j must refers the NO.i value $?(V_1,V_2,...,V_m):\{E_{-1}:E_1, E_{-2}:E_2,..., E_{-n}:E_n\}$ received(Except disobey Law A) As soon as E_{-j} is _true, it returns E_j 's value If n=1 and E_{-1} always is _true then $E_{-1}:E_1$ could write as E_1 i=1,2,...,m,j=1,2,...,n,m>0,n>0	_func
		Consist of 0-9 and at most one . and e or e- in it	A number	Float number	_num
		_nan		_nan refers nan	
		_inf		_inf refers inf	
		Consist of chars in ""	A string	"" means "	_str
		Consist of chars in ''	An error message	'' means '	_err
		_true	A bool	If E_0's value is _true then what statement expressed by E_0 is true	_bool
		_false		If E_0's value is _false then what statement expressed by E_0 is false	
		_illegal		If E_0's value is _illegal then what statement expressed by E_0 is illegal	
		_possible		If E_0's value is _possible then what statement expressed by E_0 is possible	
		{}	A list	An empty list	_list
		$\{E_1,E_2,...,E_n\}$		A list that has n(n>0) members	
		$V_0\{V_1:E_1,V_2:E_2,...,V_n:E_n\}$	An object or an error message	V_0 is a type name V_i is a member variable T is a value which type is V_0 The member variable V_{-1} of T refers 'undefined' The member variable V_i of T refers E_i's value is_V_0 is a global variable If is_V_0(T)'s value is _true then $V_0\{V_1:E_1,V_2:E_2,...,V_n:E_n\}$'s value is T else $V_0\{V_1:E_1,V_2:E_2,...,V_n:E_n\}$'s value is 'Create object error' i=1,2,...,n,n>0	
		$E_0(E_1,E_2,...,E_n)$	A function call	E_0 receives $E_1,E_2,...,E_n$ in order and return a value as $E_0(E_1,E_2,...,E_n)$'s value i=1,2,...,n,n>0	
		$E_1.V_1$	A member variable	Get the member variable V_1's value of E_1's value	
		$(\#E_1)$	A calculation	If the char before $(\#E_1)$ is (or { or , or : and the char after $(\#E_1)$ is not . or (then $(\#E_1)$ could write as $\#E_1$	
		$(E_1\#E_2)$		If the char before $(E_1\#E_2)$ is (or { or , or : and the char after $(E_1\#E_2)$ is not . or (then $(E_1\#E_2)$ could write as $E_1\#E_2$	
P_i	D_i	$V_0(N):E_0$	Define a global variable	The global variable V_0's value is E_0's value	Blank chars will be ignored except in "" or ''
		$V_0(N):$		The global variable V_0's value is inexpressible by Fx	
		$\#T_1(N):E_{-1}$	Define a calculation	If E_{-1} exists then $(\#E_1)$'s value is $E_{-1}(E_1)$'s value else if E_{-2} exists then $(\#E_1)$'s value is $E_{-2}(E_1)$'s value else $(\#E_1)$'s value is '$\#T_1$ is undefined. '	E_1's type is T_1 E_2's type is T_2
		$\#?(N):E_{-2}$			
		$T_1\#T_2(N):E_{-3}$		If E_{-3} exists then $(E_1\#E_2)$'s value is $E_{-3}(E_1,E_2)$'s value else if E_{-4} exists and E_{-5} does not exist then $(E_1\#E_2)$'s value is $E_{-4}(E_1,E_2)$'s value else if E_{-4} does not exist and E_{-5} exists then $(E_1\#E_2)$'s value is $E_{-5}(E_1,E_2)$'s value else if E_{-4} exists and E_{-5} exists then $(E_1\#E_2)$'s value is '$T_1\#T_2$ is undefined.' else if E_{-6} exists then $(E_1\#E_2)$'s value is $E_{-6}(E_1,E_2)$'s value else $(E_1\#E_2)$'s value is '$T_1\#T_2$ is undefined.'	
		$T_1\#?(N):E_{-4}$			
		$?\#T_2(N):E_{-5}$			
		$?\#?(N):E_{-6}$			
		$\$V_0$	Import the file V_0.fx	The file V_0.fx is in the standard library	V_0 is a file name
		$@V_0$		The file V_0.fx is under the same path as the file containing this $@V_0$	
	M	$D_1;D_2$	Multiple definitions or import files	$D_1;D_2$ is the same as $D_2;D_1$	