∆FIX Preprocessor Cheat Sheet

[∆FIX Cheat Sheet.md]

Cat.	Item	Example
Cont.	Continuation line symbol… or .. (dots / diaereses are ignored)	a←1 + 2 + … 3 4 5 s←’one two three .. four five’ [See also Continuing DQ Strings]
Cont.	Continuing Parenthetical Expressions Across lines	a←¯1+(2* 31 32 33)÷(1+⍳3)
Cont.	Continuing SQ Strings Across Lines	‘This is line 1 and line 2.’ ‘This is line 1 and line 2.’
Cont.	Continuing DQ Strings Across Lines	”This is line1 and line 2.” (‘This is line1’,(⎕UCS 10),’and line 2.’)
Cont.	Quotes with continuation line symbol … or ..	This is a cat.. alog. This is a cat ..alog.’ ‘This is a catalog. This is a cat alog.’
Where	Semicolon at end or beginning of line (outside parens, brackets, braces) represents ⊣ “where”. For semicolons within parens, see Lists. Parens inside brackets follow APL standards.)	S← (A×x*2)+(B×x)+C ; A←10 ; B←¯5 ; C←○1 ; x←⍳100 S←(A×x*2)+(B×x)+C⊣A←10⊣B←¯5⊣C←○1⊣x←⍳10 S 3.141592654 8.141592654 33.14159265 78.14159265 143.1415927 228.1415927 333.1415927 458.1415927 603.1415927 768.1415927
Where	What about “where” inside parentheses? Use ⊣, where you might have used ‘;’.	S←myNS.((A×x*2)+(B×x)+C ⊣ A←10 ⊣ B←¯5 ⊣ C←○1 ⊣ x←⍳100) S←myNS.((A×x*2)+(B×x)+C⊣A←10⊣B←¯5⊣C←○1⊣x←⍳10)
Unicode	Decimal ⎕Unnn	⎕U123≡⎕UCS 123 ⋄ ⎕U123≡'{'
Unicode	Hexadecimal ⎕Unhhx	⎕U7BX≡⎕U123
Nums	Hexadecimal Integers dhhhx	1122X≡123X+0FFFX
Nums	Long number separator _ (underscore)	123_245_343_122.35 3.14159_26534

∆FIX Preprocessor Cheat Sheet

[∆FIX Cheat Sheet.md]

Cat.	Item	Example
Atoms	Atoms consist of APL names, numbers, and APL strings.	:FOR a :IN `fred ‘jack 123’… 3.14159 55
Atoms	Atoms as names `atom1 atom2	colors←`red orange yellow reds← `red orange 1≡∧/reds∊colors
Atoms	Atoms as numbers	local←`CA 14850
Atoms	Atoms as strings `name1 'string2'	Last←`Smith ‘Van Buren’ Jones
Parms (Parameters)	Parameters	atom1 atom2→ arbitrary code [See Lists for examples]
Lists	Lists (code1 ; code2;)	Create mappings from names/numbers/strings to arbitrary code expressions
Lists	Ordinary code (code1; code2;)	test←(⍳3 ; ⍳4)
Lists	Function parameters	graph←(XY type 3→(⍳20)(1○⍳20); legend x→’Voltage’; legend y→’Amplitude’)
Lists	With atoms	graph(type→`XY 3; smooth → `true; line color→`green; line height→`2.5 in)
Lists	Omitted parameters (code1;;code3)	address(2525; ‘Cozy’; ‘Lane’; ; ; 90212; USA) ⍝ city/state opal with zip
Lists	Null list ()	Always true: () ≡ ⍬
Name Suffixes	Is name defined? name..DEF	:IF print..DEF
Name Suffixes	Is name undefined? name..UNDEF	:IF print..UNDEF
Name Suffixes	Put name in quotes: name..Q (possibly after macro or other processing)	⎕NPARTS fileName..Q
Name Suffixes	Get value of environment variable ‘name'	PATH←PATH..ENV{×≢⍺: ⍵ <> ⍺}’.:..’
Directive	If clause ::IF code	::IF 0≠≢DEBUG..ENV
Directive	Test that name is defined	::IFDEF DEBUG_FLAG
Directive	Test that name is not defined	::IFNDEF DEBUG_FLAG
Directive	Undefine name	::UNDEF DEBUG_FLAG

Cat.	Item	Example
Directive	Else-if clause ::ELSEIF/ELIF code	::ELSEIF DEBUG_FLAG≥3
Directive	Terminate ::IF, ::IFDEF or ::IFNDEF sequence	::END, ::ENDIF, ::ENDIFDEF, ::ENDIFNDEF
Directive	Conditional with single variable name	::COND DEBUG ⎕←CUR_RESULT
Directive	Conditional with arbitrary parenthetical expression	::COND (DEBUG≥3) ⎕←CUR_RESULT
Directive	Preprocessor messages ::MESSAGE/MSG text	::MSG DEBUGGER CODE ACTIVATED!
Directive	Preprocessor error msgs ::ERROR [num] string	::IF CONFLICTING_OPTIONS ::ERROR 911 Conflicting Options Detected!
Directive	Include a file unconditionally	::INCLUDE MyLocalData.dat
Directive	Include a file if not already included earlier	::CINCLUDE printServices.dyalog