

# Mu Runtime Reference

version 0.2.13

## type keywords and aliases

<i>supertype</i>	<i>T</i>
<i>bool</i>	<i>()</i> , <i>:nil</i> are false, otherwise true
<i>condition</i>	<i>keyword</i> , see <b>exceptions</b>
<i>list</i>	<i>:cons</i> or <i>()</i> , <i>:nil</i>
<i>ns</i>	<i>#\$(:ns #(:t fixnum symbol))</i>
<i>ns-designator</i>	<i>ns</i> , <i>:nil</i> , <i>:unqual</i>
<i>:null</i>	<i>()</i> , <i>:nil</i>
<i>:char</i>	<i>char</i>
<i>:cons</i>	<i>cons</i> , <i>list</i>
<i>:fixnum</i>	<i>fixnum</i> , <i>fix</i>
<i>:float</i>	<i>float</i> , <i>fl</i>
<i>:func</i>	<i>function</i> , <i>fn</i>
<i>:keyword</i>	<i>keyword</i> , <i>key</i>
<i>:stream</i>	<i>stream</i>
<i>:struct</i>	<i>struct</i>
<i>:symbol</i>	<i>symbol</i> , <i>sym</i>
<i>:vector</i>	<i>vector</i> , <i>string</i> , <i>str</i>
	<i>:bit</i> <i>:char</i> <i>:t</i>
	<i>:byte</i> <i>:fixnum</i> <i>:float</i>

## core

<i>apply fn list</i>	<i>T</i>	apply <i>fn</i> to <i>list</i>
<i>compile form</i>	<i>T</i>	<i>mu</i> form compiler
<i>eq T T'</i>	<i>bool</i>	<i>T</i> and <i>T'</i> identical?
<i>eval form</i>	<i>T</i>	evaluate <i>form</i>
<i>type-of T</i>	<i>key</i>	type keyword
<i>view for</i>	<i>vector</i>	vector of object
<i>fix fn T</i>	<i>T</i>	fixpoint of <i>fn</i>
<i>gc</i>	<i>bool</i>	garbage collection
<i>repr T</i>	<i>vector</i>	tag representation
<i>unrepr vector</i>	<i>T</i>	tag representation

## special forms

<i>:lambda list . list'</i>	<i>function</i>	anonymous <i>fn</i>
<i>:alambda list . list'</i>	<i>function</i>	anonymous <i>fn</i>
<i>:quote T</i>	<i>list</i>	quoted form
	<i>T</i>	conditional

*vector* is an 8 element *:byte* vector of little-endian argument tag bits.

<b>frames</b>			<b>vectors</b>		
		frame binding: <i>(fn . #(:t ...))</i>			
<i>%frame-stack</i>	<i>list</i>	active frames	<i>make-vector</i>	<i>key list</i>	specialized vector from list
<i>%frame-pop fn</i>	<i>frame</i>	pop function's top frame binding	<i>vector-length</i>	<i>vector</i>	length of vector
<i>%frame-push frame</i>	<i>cons</i>	push frame	<i>vector-type</i>	<i>vector</i>	type of vector
<i>%frame-ref fn fix</i>	<i>T</i>	function, offset	<i>sref</i>	<i>vector fix</i>	<i>n</i> th element
<b>symbols</b>			<b>namespaces</b>		
<i>boundp sym</i>	<i>bool</i>	is symbol bound?	runtime namespaces: <i>mu</i> (static), <i>keyword</i>		
<i>make-symbol string</i>	<i>sym</i>	uninterned symbol	<i>make-namespace</i>	<i>str</i>	make namespace
<i>symbol-namespace sym</i>	<i>ns-designator</i>	namespace designator	<i>namespace-name</i>	<i>ns</i>	namespace name
<i>symbol-name symbol</i>	<i>string</i>	name binding	<i>intern ns str value</i>	<i>symbol</i>	intern symbol in non-static namespace
<i>symbol-value symbol</i>	<i>T</i>	value binding	<i>find-namespace</i>	<i>str</i>	map string to namespace
<b>fixnums</b>			<i>find ns string</i>	<i>symbol</i>	map string to symbol
<i>add fix fix'</i>	<i>fixnum</i>	sum	<b>structs</b>		
<i>ash fix fix'</i>	<i>fixnum</i>	arithmetic shift	<i>make-struct</i>	<i>key list</i>	type key from list
<i>div fix fix'</i>	<i>fixnum</i>	quotient	<i>struct-type</i>	<i>struct</i>	struct type key
<i>less-than fix fix'</i>	<i>bool</i>	<i>fix &lt; fix'?</i>	<i>struct-vec</i>	<i>vector</i>	of struct members
<i>logand fix fix'</i>	<i>fixnum</i>	bitwise and	<b>streams</b>		
<i>lognot fix</i>	<i>fixnum</i>	bitwise complement	<i>*standard-input*</i>	<i>stream</i>	std input stream
<i>logor fix fix'</i>	<i>fixnum</i>	bitwise or	<i>*standard-output*</i>	<i>stream</i>	std out stream
<i>mul fix fix'</i>	<i>fixnum</i>	product	<i>*error-output*</i>	<i>stream</i>	std error stream
<i>sub fix fix'</i>	<i>fixnum</i>	difference	<i>open type dir str bool</i>	<i>stream</i>	open stream, raise error if bool
<b>floats</b>			<i>type dir :file :input</i>	<i>:string output</i>	<i>:bidir</i>
<i>fadd fl fl'</i>	<i>float</i>	sum	<i>close stream</i>	<i>bool</i>	close stream
<i>fdiv fl fl'</i>	<i>float</i>	quotient	<i>openp stream</i>	<i>bool</i>	is stream open?
<i>fless-than fl fl'</i>	<i>bool</i>	<i>fl &lt; fl'?</i>	<i>flush stream</i>	<i>bool</i>	flush stream
<i>fmul fl fl'</i>	<i>float</i>	product	<i>get-string stream</i>	<i>string</i>	from string stream
<i>fsub fl fl'</i>	<i>float</i>	difference	<i>read-byte stream bool T</i>	<i>byte</i>	read byte from stream, error on eof, <i>T</i> : eof-value
<b>conses/lists</b>			<i>read-char stream bool T</i>	<i>char</i>	read char from stream, error on eof, <i>T</i> : eof-value
<i>append list</i>	<i>list</i>	append lists	<i>unread-char char stream char</i>		push char onto stream
<i>car list</i>	<i>T</i>	head of list	<i>write-byte byte stream</i>	<i>byte</i>	write byte
<i>cdr list</i>	<i>T</i>	tail of list	<i>write-char char stream</i>	<i>char</i>	write char
<i>cons T T'</i>	<i>cons</i>	( <i>T</i> , <i>T'</i> )	<i>read stream bool T</i>	<i>T</i>	read stream
<i>length list</i>	<i>fixnum</i>	length of list	<i>write T bool stream</i>	<i>T</i>	write with escape
<i>nth fix list</i>	<i>T</i>	<i>nth car of list</i>			
<i>nthcdr fix list</i>	<i>T</i>	<i>nth cdr of list</i>			

