Emacs Lisp Types

Main Type Category	Sub-category	Sub-category	Sub-category	References
Last updated on:	2026-01-21			
Symbols	**Symbols**			• **Symbol Type** • **Symbol components** • **Creating and interning symbols**
	booleans			
	comparison			• **Equality predicates** • **Comparison functions @ Emacs Wiki**
Numbers	**Integers**			
	Floats			
	complex-numbers			Complex numbers are not explicitly supported by Emacs Lisp. • The **cplx external library** supports the concept of complex numbers.
Characters and Strings	Char			• **Basic char literal syntax**
	Strings			• **Creating Strings** • **String Modifications @ Emacs Wiki** • **Replace in String @ Emacs Wiki** • **Split String @ Emacs Wiki** • **String trim @ ∑Xah**
• **Encodings**				• **Unicode Encoding @ Emacs Wiki** • **Emacs Unicode Pitfalls,** by Christopher Wellons, 2014-06-13
Ordered Collection of elements See also: • **Pure storage** • **Elisp Cookbook @ Emacs Wiki**	**Sequence**			• **Sequence Functions**
		List		• **List Modifications @ Emacs Wiki** • **List Destructive Operations @ Emacs Wiki** • **Sorting Lists**
			cons cell	Note: a cons cell is not a sequence. The cons cell is placed here because of its strong relationship with lists.
			alist - association list	• **Elisp: Association List @ ∑Xah**
			cl association list	
			plist - property list	• **Elisp: Property List @ ∑Xah** • **Alist Vs Plist @ Emacs Wiki**
			set - (lists as sets)	
		Array		• **Array Functions**
			Vector	• **Vector Functions**
			Bool-vector	
			Char-table	
			Ring	
			String	See strings above. Strings are sequences of characters.
Creation of New Object Types	**Record**			Used as underlying representations of *cl-defstruct* and *defclass* instances.
	Structure			• **Options for Structured Data in Emacs Lisp**, by Christopher Wellons, 2018-02-14 • **The Common Lisp Cookbook** - **Data Structures** - **Structures**
	Hash-table			
ieieo - CLOS for Emacs Lisp	**classes**			
Emacs Specialized Types	**Buffers**			
	Markers			• **Functions that create markers**
	Overlays			
	Process			• **Create synchronous process** • **Create asynchronous process** • **Reddit discussion about async process with Eli Zaretskii** • Also read: **Command Loop**
Functions				• **Emacs Lisp Code Guidelines - Functions, lambdas, macros**
	functions			**Defining a function** with: • **defun macro** • **function argument list of functions defined with the defun macro** • **inline functions** • **Calling functions indirectly** • **cl-lib macros: cl-defun, cl-function, cl-defsubst, …** • See also Common Lisp references (which explain what Emacs Lisp cl-lib emulates): • **The Common Lisp Cookbook** – **Functions** • **Practical Common Lisp** - **Common Lisp functions** • Note that Emacs Lisp cl-defun support *&aux auxiliary variables* in argument list, something not available in Common Lisp.
	advices • **Ways to compose advice**			• **Advice @ Wikipedia** • **The Limits of Emacs Advice**, by Christopher Wellons, 2013-01-22 • **Advising Functions @ Emacs Wiki** • **Advice Vs Hooks @ Emacs Wiki** • Meta-Programming in Emacs Using defadvice, T.V. Raman, 2019-10-16
	lambda			• **Anonymous functions** • **lambda expressions** • **What's in an Emacs lambda**, by Christopher Wellons, 2017-12-14 • **Emacs Lisp Lambda Expressions Are Not Self-Evaluating**, by Christopher Wellons, 2018-02-22
	closures (lambda)			• **Emacs Lisp Readable Closures**, by Christopher Wellons, 2013-12-30 • **Lexical binding**
	generators (and iterators)			• **Emacs 25.1 bring Generators,** by Christopher Wellons, 2018-05-31
ieieo - CLOS for Emacs Lisp	**methods**			• **Generic functions**