

core.async essential functions

Function	Doco
<!	Usage: (<! port) takes a val from port. Must be called inside a (go ...) block. Will return nil if closed. Will park if nothing is available.
>!	Usage: (>! port val) puts a val into port. nil values are not allowed. Must be called inside a (go ...) block. Will park if no buffer space is available. Returns true unless port is already closed.
<!!	Usage: (<!! port) takes a val from port. Will return nil if closed. Will block if nothing is available.
>!!	Usage: (>!! port val) puts a val into port. nil values are not allowed. Will block if no buffer space is available. Returns true unless port is already closed.
timeout	Usage: (timeout msecs) Returns a channel that will close after msecs
chan	Usage: (chan) (chan buf-or-n) Creates a channel with an optional buffer. If buf-or-n is a number, will create and use a fixed buffer of that size.
alt!	Usage: (alt! & clauses) Makes a single choice between one of several channel operations, as if by alts!, returning the value of the result expr corresponding to the operation completed. Must be called inside a (go ...) block.
go	Usage: (go & body) Asynchronously executes the body, returning immediately to the calling thread. Additionally, any visible calls to <!, >! and alt!/alts! channel operations within the body will block (if necessary) by 'parking' the calling thread rather than tying up an OS thread (or the only JS thread when in ClojureScript). Upon completion of the operation, the body will be resumed. Returns a

	channel which will receive the result of the body when completed
close!	Usage: (close! chan) Closes a channel. The channel will no longer accept any puts (they will be ignored). Data in the channel remains available for taking, until exhausted, after which takes will return nil. If there are any pending takes, they will be dispatched with nil. Closing a closed channel is a no-op. Returns nil.