

U5ZAJ15P0 : ah
U051SS2EU : I guess I should have explicitly mentioned the `~` idiom to prevent namespacing
U5ZAJ15P0 : is there a way to do the opposite? selectively pick the symbols I want to namespace qualify
U051SS2EU : that's harder
U5ZAJ15P0 : instead of the ones I do not want to namespace qualify
U5ZAJ15P0 : mmh ok
U5ZAJ15P0 : nevermind then
U5ZAJ15P0 : thanks :slightly_smiling_face:
U051SS2EU : this is close, but no cigar ```peregrine.circle=> [:find ... :where [^identity ?foo]][:find ... :where [(quote clojure.core/identity) ?foo]]```\n\nU5ZAJ15P0 : yep, not quite :confused:
U5ZAJ15P0 : why wouldn't this work:```\n[:find ... :where [~^identity ?foo]]\n```\n\n?\n\nU051SS2EU : ~ doesn't make sense outside `
U5ZAJ15P0 : why does it make sense here:```\n[:find ... :where [identity ~'?foo]]\n```\n\n?\n\nU051SS2EU : it's inside the ` at the beginning of the vector
U051HUZLD : thanks
U5ZAJ15P0 : right, but why does swapping out the use of syntax quoting vs normal quoting makes it different in use?
U5ZAJ15P0 : (I'm not familiar enough with quoting yet)
U051SS2EU : because ~ is only valid in syntax quote
U5ZAJ15P0 : oh
U051SS2EU : it's a special operation that is only defined in that context
U5ZAJ15P0 : using it outside throws ```java.lang.IllegalStateException: Attempting to call unbound fn: #clojure.core/unquote```\nU5ZAJ15P0 : does this only get bound within a syntax quote?