What is a seq?

What makes a collection seq-able?

How can you get an actual seq from a seq-able?

How do you move through a seq?

Functions on collections result in ...

Where are custom seqs defined for most seq-ables?

How can maps be viewed as seqs?

How can you get a reliable order of traversal
for maps and sets?

Describe `conj` and `into`.

All Clojure sequences are ... and most are ...

What's an easy way to insert commas
between words?

What's the difference between `set`/`vec` and `hash-set`/`vector`?

What are the common sequence factory methods that take a variable number of elements as arguments?

What are the common sequence filtering
functions?

How can you create a new predicate that reverses the truth conditions of an existing one?

What are the common sequence predicates?

What are the common sequence transforming
functions?

Give the signature for seq comprehensions.

How are they implemented?

Which filter expressions are available for seq
comprehensions?

Force the evaluation of a lazy sequence.

What can be treated as a seq, Clojure
collections aside?

Which structure-specific functions do lists
support?

Which structure-specific functions do vectors support?

What's the difference between `take/drop` and `subvec`?

Which structure-specific functions do maps
support for querying contents?

What's the trouble with looking up a key in a map?

How can you avoid this problem?

What structure-specific functions do maps
support for creating new maps?

Unlike other structure-specific functions, those for sets must be ...

Which set-theoretic operations do sets
support?

How can database results be viewed as seq-able.

Which database-like functions does Clojure support?