```
%YAML 1.1   # Reference card
---
Collection indicators:
  '? ' : Key indicator.
  ': ' : Value indicator.
  '- ' : Nested series entry indicator.
  ', ' : Separate in-line branch entries.
  '[]' : Surround in-line series branch.
  '{}' : Surround in-line keyed branch.
Scalar indicators:
  ''' : Surround in-line unescaped scalar ('' escaped ').
  '"'  : Surround in-line escaped scalar (see escape codes below).
  '|'  : Block scalar indicator.
  '>'  : Folded scalar indicator.
  '-'  : Strip chomp modifier ('|-' or '>-').
  '+'  : Keep chomp modifier ('|+' or '>+').
  1-9  : Explicit indentation modifier ('|1' or '>2').
         # Modifiers can be combined ('|2-', '>+1').
Alias indicators:
  '&'  : Anchor property.
  '*'  : Alias indicator.
Tag property: # Usually unspecified.
  none    : Unspecified tag (automatically resolved by application).
  '!'     : Non-specific tag (by default, "!!map"/"!!seq"/"!!str").
  '!foo'  : Primary (by convention, means a local "!foo" tag).
  '!!foo' : Secondary (by convention, means "tag:yaml.org,2002:foo").
  '!h!foo': Requires "%TAG !h! <prefix>" (and then means "<prefix>foo").
  '!<foo>': Verbatim tag (always means "foo").
Document indicators:
  '%'  : Directive indicator.
  '---': Document header.
  '...': Document terminator.
Misc indicators:
  ' #' : Throwaway comment indicator.
  '`@' : Both reserved for future use.
Special keys:
  '='  : Default "value" mapping key.
  '<<' : Merge keys from another mapping.
Core types: # Default automatic tags.
  '!!map' : { Hash table, dictionary, mapping }
  '!!seq' : { List, array, tuple, vector, sequence }
  '!!str' : Unicode string
More types:
  '!!set' : { cherries, plums, apples }
  '!!omap': [ one: 1, two: 2 ]
Language Independent Scalar types:
  { ~, null }              : Null (no value).
  [ 1234, 0x4D2, 02333 ]   : [ Decimal int, Hexadecimal int, Octal int ]
  [ 1_230.15, 12.3015e+02 ]: [ Fixed float, Exponential float ]
  [ .inf, -.Inf, .NAN ]    : [ Infinity (float), Negative, Not a number ]
  { Y, true, Yes, ON  }    : Boolean true
  { n, FALSE, No, off }    : Boolean false
  ? !!binary >
      R0lG...BADS=
  : >-
      Base 64 binary value.
Escape codes:
 Numeric   : { "\x12": 8-bit, "\u1234": 16-bit, "\U00102030": 32-bit }
 Protective: { "\\": '\', "\"": '"', "\ ": ' ', "\<TAB>": TAB }
 C         : { "\0": NUL, "\a": BEL, "\b": BS, "\f": FF, "\n": LF, "\r": CR,
             "\t": TAB, "\v": VTAB }
 Additional: { "\e": ESC, "\_": NBSP, "\N": NEL, "\L": LS, "\P": PS }
...
```