

;; Produces a true or false answer depending on if the label
appears within three words of the name

```
(define (label-near? label name word-one word-two word-three)
  (cond [(and (string=? "name" "word-one")
               (string=? "label" "word-two") "true")]
        [(and (string=? "name" "word-one")
               (string=? "label" "word-three") "true")]
        [(and (string=? "name" "word-two")
               (string=? "label" "word-one") "true")]
        [(and (string=? "name" "word-two")
               (string=? "label" "word-three") "true")]
        [else "false"]])
```

Annotations:

- Red: points to the `(cond` keyword.
- Green: points to the first clause `[(and (string=? "name" "word-one") (string=? "label" "word-two") "true")]`.
- Blue: points to the first clause `[(and (string=? "name" "word-one") (string=? "label" "word-two") "true")]`.
- Green: points to the closing bracket of the first clause `]`.
- Red: points to the `[else "false"]` clause.

Welcome to [DrScheme](#), version 4.2.2 [3m].

Language: Beginning Student; memory limit: 128 megabytes.

`cond:` expected a clause with a question and answer, but found a
clause with only one part

Annotations:

- Blue: points to the error message `clause with only one part`.
- Green: points to the `>` prompt.
- Red: points to the error message `expected a clause with a question and answer, but found a`.