Read

```
CLIPS> (clear)
CLIPS> (defrule read-input
           (initial-fact)
        =>
           (printout t "Name a primary color " crlf)
           (assert (color (read))))
CLIPS> (defrule check-input
            ?color <- (color ?color-read&red|yellow|blue)
         =>
            (retract ?color)
            (printout t "Correct" crlf))
CLIPS> (reset)
CLIPS> (run)
Name a primary color
red
Correct
CLIPS> (reset)
CLIPS> (run)
Name a primary color
green
CLIPS>
```

readline for multiple arguments

```
CLIPS> (clear)
CLIPS> (assert-string "(primary color is red)")
<Fact-0>
CLIPS> (facts)
f-0     (primary color is red) CF 1.00
For a total of 1 fact.

CLIPS> (clear)
CLIPS> (defrule test-readline
          (initial-fact)
         =>
          (printout t "Enter Input" crlf)
          (bind ?string (readline))
          (assert-string (str-cat "(" ?string ")")))
CLIPS> (reset)
CLIPS> (run)
Enter Input
primary color is red
CLIPS> (facts)
f-0     (initial-fact) CF 1.00
f-1     (primary color is red) CF 1.00
For a total of 2 facts.
CLIPS>
```