

```
-- Week 1, Activity 14: Predicates
-- Predicates are a way to "look ahead" on a subject,
-- without consuming that part of subject.
-- The principal predicate that we'll use is the
-- NOT predicate: - (unary minus operator).

local lpeg = require "lpeg"

-- The NOT predicate matches when the pattern that it is
-- applied to doesn't match.
p = -lpeg.P("a")
print(p:match("hello"))      --> 1 (doesn't consume anything)
print(p:match("aaa"))        --> nil

-- Suppose that "do" is a reserved word. How can I identify
-- "do" as a word? This kind of thing is very useful to
-- distinguish reserved words from parts of identifiers.
letra = lpeg.R("az", "AZ")
p = lpeg.P("do") * -letra
print(p:match("do"))          --> 3
print(p:match("do "))         --> 3
print(p:match("do()"))        --> 3
print(p:match("don"))         --> nil

-- Other interesting use of the not predicate is the following
-- pattern, that means NOT matches any character, that is,
-- matches only the empty string:
p = -lpeg.P(1)
print(p:match(""))            --> 1
print(p:match("a"))           --> nil

-- If we put this pattern at the end, for example, of a
-- pattern to identify "do", it matches only when "do"
-- is at the end of the string:
p = lpeg.P("do") * -lpeg.P(1)
print(p:match("do"))          --> 3
print(p:match("do "))         --> nil

-- Other useful trick is put any negative number as
-- argument to 'P', it matches only when there are not
-- that many characters in the string. It's another
-- way to write "-lpeg.P(1)":
p = lpeg.P(-1)
print(p:match(""))            --> 1
print(p:match(" "))           --> nil
```