

# Lexer rules

`name` : ... ; : rule definition

(...) : grouping

| : alternative, e.g. (a|b)

's' : char or string literal. '\n' for newline.

. : any character

a .. b : range, e.g. ('0'..'9')

{...} : action

+ : 1 or more, e.g. ('0' .. '9')+

\* : 0 or more

? : optional (or semantic predicate)

[...] : choice between characters, e.g. [abc]

~[...] : match not, e.g. ~[abc]

// ... : single-line comment

/\* ... \*/ : multi-line comment

# Parser rules

```
parser grammar explParser;  
  
// extern tokens definition  
options {tokenVocab=explLexer;}  
  
// Parsing rules: structure tokens together  
prog : (anbn | cndn) EOF ;  
// EOF: end-of-file token (predefined)  
  
anbn : A anbn B  
      | /* nothing for empty alternative */  
      ;  
  
cndn : 'C' cndn 'D' |;  
  
// Compile/execute with:  
// antlr4 explLexer.g4 explParser.g4  
// javac *.java  
// echo 'aabb' | grun expl prog -gui
```