

TD - Deuxième semestre informatique

Pierre-Etienne Moreau



Préparation du TD

Nous utilisons l'outil `lark` (<https://github.com/lark-parser/lark>) pour décrire des grammaires et pour obtenir un analyseur syntaxique pour ces grammaires.

Pour préparer le TD vous devez installer l'outil sur votre machine

Pour cela vous devez exécuter la commande suivante :

```
pip install lark
```

Pour tester que votre installation fonctionne bien vous pouvez exécuter le programme

Python suivant :

```
import lark
GRAMMAR = """
?start: program

program: line+

?line: linenumber command -> line
?linenumber: INT

?command: COMMENT                                -> comment
        | "LET" variable "=" expression          -> let
        | "GOTO" linenumber                       -> goto
        | "PRINT" expression                      -> print
        | "INPUT" variable                        -> input
        | "IF" expression "THEN" linenumber       -> if

?variable: CNAME
?expression: NUMBER                                -> num
        | CNAME                                    -> var
        | ESCAPED_STRING                          -> string
        | UNARY_OP expression                     -> unary
        | expression BINARY_OP expression         -> binary
        | "(" expression ")"
```

```

COMMENT: "REM" / [^\n]*/x
UNARY_OP: "-" | "!"
BINARY_OP: "+" | "-" | "*" | "/" | "%"
          | "=" | "<" | ">"
          | "&" | "|"

```

```

%import common.INT
%import common.NUMBER
%import common.WS
%import common.CNAME
%import common.ESCAPED_STRING
%ignore WS
"".strip()

```

```

parser = lark.Lark(GRAMMAR)

```

```

p1 = ""
10 LET x = 3
20 PRINT x
""

```

```

tree = parser.parse(p1)

```

```

print(tree)

```

Vous devez voir le résultat suivant :

```

Tree(Token('RULE', 'program'), [Tree('line', [Token('INT', '10'),
Tree('let', [Token('CNAME', 'x'), Tree('num', [Token('NUMBER',
'3'])])])]), Tree('line', [Token('INT', '20'), Tree('print',
[Tree('var', [Token('CNAME', 'x')])])])])])

```