

# Automi e Linguaggi Formali

a.a. 2017/2018

LT in Informatica  
26 Aprile 2018



UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA

- Il laboratorio può essere svolto in gruppi di **massimo due persone**
- Permette di ottenere un bonus di **massimo 2 punti**
- Testo e consegna degli esercizi **sul moodle** del corso
- Uno solo dei componenti consegna gli esercizi, indicando i **nomi dei componenti del gruppo**
- Gli esercizi vanno consegnati entro le **ore 23:55 di venerdì 1 Giugno 2018**

```
begin
  let a be 5
  let b be 10
  add 3 to b
  add b to a
  add a to b
  print b
  print 3
end
```

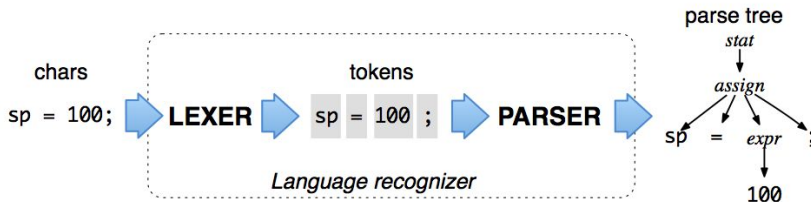
```
begin
  let a be 5
  let b be 10
  add 3 to b
  add b to a
  add a to b
  print b
  print 3
end
```

**Traduttore**

```
#include <iostream>

using namespace std;

int main() {
  int a = 5;
  int b = 10;
  b += 3;
  a += b;
  b += a;
  cout << b << endl;
  cout << 3 << endl;
}
```



- Il lexer prepara i **token** da passare al parser
- Il parser accetta uno stream dei token preparati dal lexer e cerca di riconoscere la **struttura del testo**
- La struttura del testo viene rappresentata con un **albero sintattico**

- Strumento automatico che permette di generare **analizzatori sintattici** di testi (**parser**)
- Sviluppato in Java permette di generare codice in **C++**, Java, C#, Python, Swift, Go, ...
- Supporta **espressioni regolari**, **grammatiche EBNF** e la generazione di **alberi sintattici**



Questo strumento può produrre **in automatico** il codice relativo a:

- **Lexer**: parte lessicale dove si definiscono i token e i separatori da ignorare
- **Parser**: parte grammaticale contenente le regole di produzione
- **Listener**: parte semantica che definisce le azioni da eseguire sul testo