Objectives

- SLogo Design
  - Parsing commands

Review

- How is a programming language processed?

More Sophisticated Example

- `jelan.elan.*`
  - Breaks classes into appropriate packages: Tokens, Expressions, Instructions, Parsers
- `jelan.elan.instruction`
  - Represent instructions
  - `evaluate` method
  - Check out `Print` class
- `jelan.elan.parser`
  - Parse Tokens to create Instructions
  - Check out `PrintParser` class

More Sophisticated Example

- `jelan.elan.*`
- Mapping between `Token, Instruction, Parser, instructions.prop`
  - Knows which Parser to call based on instructions.prop and mapping from Token to Parser
- Run `ElanParser` with `tests/assign_repeat`

Comparing ElanParsers

- StreamTokenizers
- parse() method
- nextToken() method

Practice Adding Instructions

- Create a token for instruction (probably a subclass of token.ReservedToken)
  - Same prefix as new instruction, e.g., IfToken.java
- Create a parser for the instruction with same prefix as instruction, e.g., IfParser.java
  - Parsing class (presumably implementing Parser) returns an instance of parsed Instruction
- Create an instruction with prefix name, e.g., If.java
- Add instruction name to file instructions.prop, e.g., add a single line to file containing string If

What Do You Think You'll Need To Do About Commands/Instructions?