Interpreter: Interpreters are all around us. Even now, in this very room.

1. Textual input need to be processed
   1. E.g. turned into OOP structures
2. Some examples
   1. Programming language compilers, interpreters and IDEs
   2. HTML, XML and similar
   3. Numeric expressions (3+4/5)
   4. Regular expressions
3. Turning strings into OOP based structures in a complicated process
4. Interpreter: A component that processes structured text data. Does so by turning it into separate lexical tokens (lexing) and then interpreting sequences of said tokens (parsing).

Handmade interpreter: Lexing & Parsing

1. You have a String input like “(13+4)-(12+1)”.
2. Lexing gets the components separately.
3. Parsing combines these components and evaluates the result in an OOP manner.
4. Example code!! – In the video

Antlr

1. Tool for interpretation
2. Check it out

Summary

1. Baring simple cases, an interpreter acts in two stages
2. Lexing turns text into a set of tokens, e.g. 3\*(4+5) → lit[3] Star LParen Lit[4] Plus Lit[]5 RParen
3. Parsing tokens into meaningful constructs → MultiplicationExpression[Integere[3], AdditionExpression[Integer[4], Integer[5]]]
4. Parsed data can then be traversed.