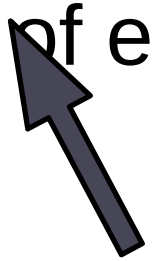
What is Parsing?

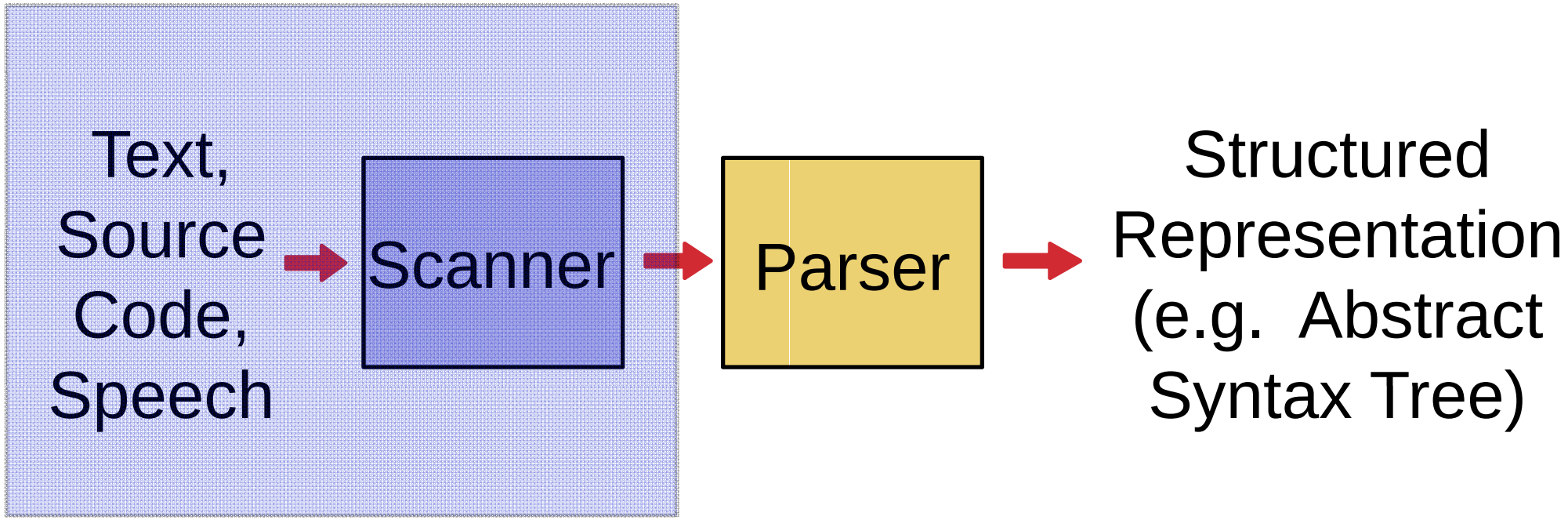
- **parse** (*v.*) to break down into its component parts of speech with an explanation of the form, function, and syntactical relationship of each part. *--The American Heritage Dictionary*



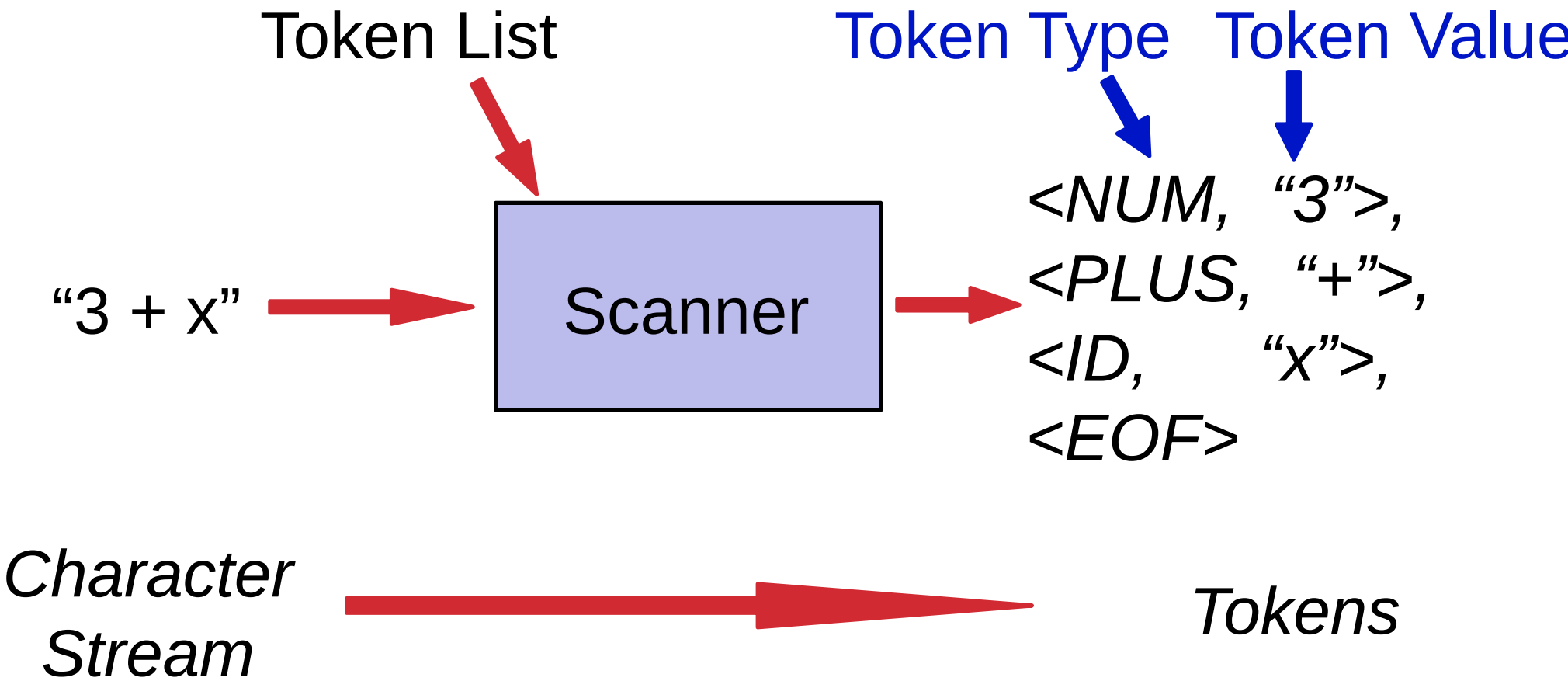
Overview

- *Intro to scanning*
- *Intro to parsing*
- *Basics of building a scanner in Java*
- Lab: *Implementing a scanner*
- *Basic Parsing Theory*
- *Design of an Object-Oriented Parser*

High-Level View



Scanning: Breaking Things Down



Scanning: Token List

Tolkien List

Token List

Token Type

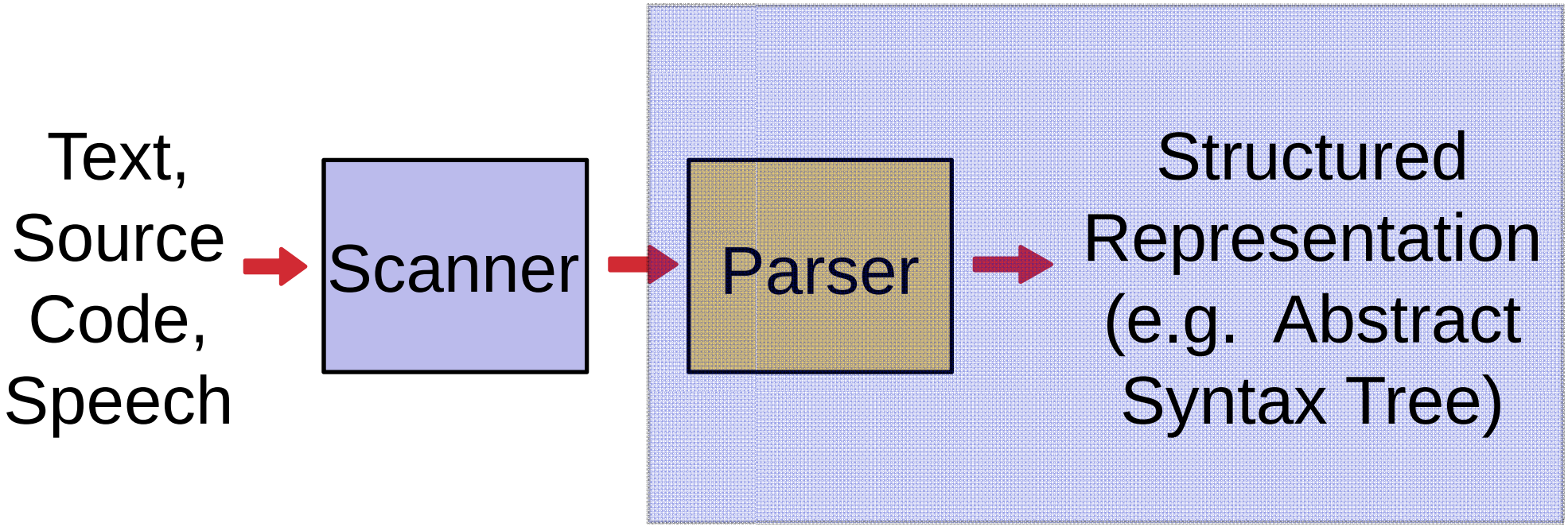
Token Descriptor

- <*NUM*: [(”0” - “9”)+]>
- <OP: [”+”, “*”]>
- <*ID*: [alpha (alphaNum)*]>

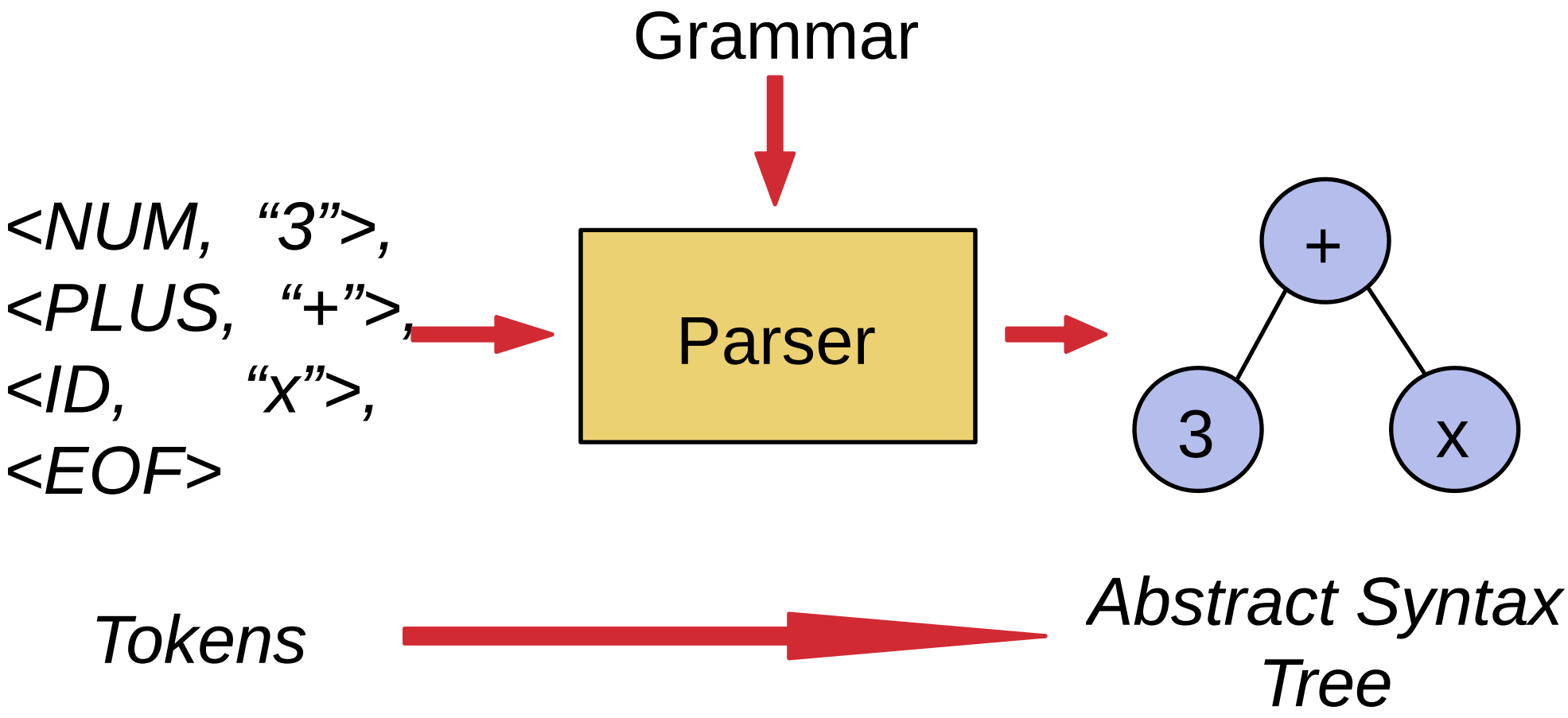
Tolkien Descriptor = *Magical Powers*

Tolkien Type = *Wizard*

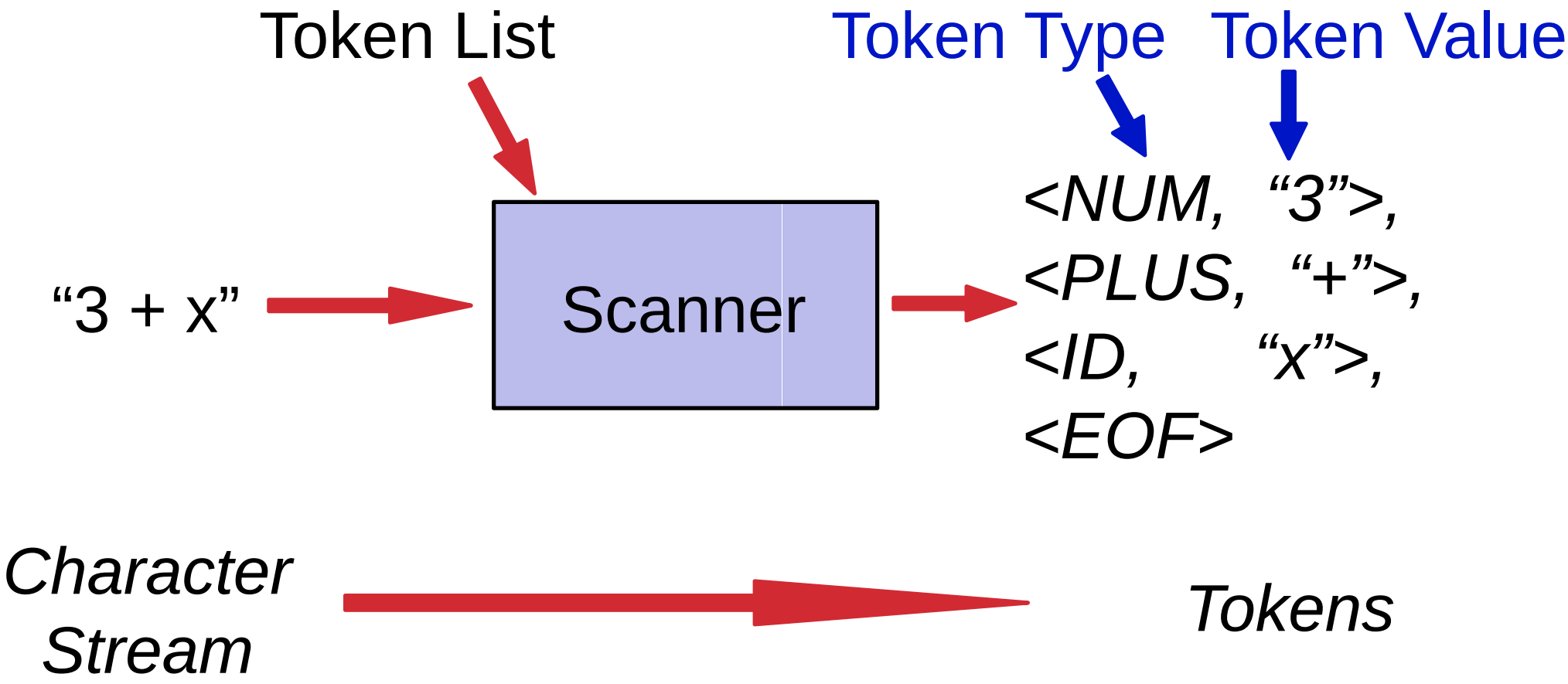
High-Level View (you saw this earlier)



Parsing: Organizing Things



Manual Scanning in Java



Tokenizing Example

- public static final int PLUS = '+';

Initialization

- public void tokenize(String str) {
  - StreamTokenizer stok = new StreamTokenizer(new StringReader(str));
  - int token;

Configuration

  - stok.ordinaryChar(PLUS);
  - stok.parseNumbers();

  - while((token = stok.nextToken()) != StreamTokenizer.TT_EOF) {
    - switch(token) {
    - case TT_WORD:
      - System.out.println("WORD = " + stok.sval); break;
    - case TT_NUMBER:
      - System.out.println("NUM = " + stok.nval); break;
    - case PLUS:
      - System.out.println("PLUS"); break;
    - }
  - }
- }

Scanning

Tokenizing Example

- public static final int PLUS = ‘+’;

Initialization

- public void tokenize(String str) {
  - StreamTokenizer stok = new StreamTokenizer(new StringReader(str));
  - int token;

Configuration

  - stok.ordinaryChar(PLUS);
  - stok.parseNumbers();

  - while((token = stok.nextToken()) != StreamTokenizer.TT_EOF) {
    - switch(token) {
    - case TT_WORD:
      - System.out.println(”WORD = “ + stok.sval); break;
    - case TT_NUMBER:
      - System.out.println(”NUM = “ + stok.nval); break;
    - case PLUS:
      - System.out.printlN(”PLUS”); break;
    - }
  - }
- }

Scanning

java.io.StreamTokenizer

Configuration: It's like programming a VCR!

Token Type (int)	*Token Desc.*	*How to Customize*
StreamTokenizer.TT_WORD	a word (no spaces)	void wordChars(int low, int high)
int qch e.g. *:hello there:* is a quoted string	a string quoted by 'qch'	void quoteChar(int qch) e.g. quoteChar(':')
StreamTokenizer.TT_NUMBER	Numbers	void parseNumbers()
int ch e.g. (int) '+'	the character value of *ch*	void ordinaryChar(int ch) e.g. ordinaryChar('+')

Tokenizing Example

- public static final int PLUS = '+';

Initialization

- public void tokenize(String str) {
  - StreamTokenizer stok = new StreamTokenizer(new StringReader(str));
  - int token;

Configuration

  - stok.ordinaryChar(PLUS);
  - stok.parseNumbers();

  - while((token = stok.nextToken()) != StreamTokenizer.TT_EOF) {
    - switch(token) {
    - case TT_WORD:
      - System.out.println("WORD = " + stok.sval); break;
    - case TT_NUMBER:
      - System.out.println("NUM = " + stok.nval); break;
    - case PLUS:
      - System.out.println("PLUS"); break;
    - }
  - }
- }

Scanning

java.io.StreamTokenizer

Scanning: It's like calling "nextToken" over and over!

- Call int StreamTokenizer.nextToken() to get the next token.
  - String StreamTokenizer.sval (public field) holds the token value for TT_WORD and quote token types
  - double StreamTokenizer.nval (public field) holds the token value for TT_NUMBER token type

Tokenizing Example

- public static final int PLUS = ‘+’;
- public void tokenize(String str) {
  - StreamTokenizer stok = new StreamTokenizer(new StringReader(str));
  - int token;
  - stok.ordinaryChar(PLUS);
  - stok.parseNumbers();
  - while((token = stok.nextToken()) != StreamTokenizer.TT_EOF) {
    - switch(token) {
    - case TT_WORD:
      - System.out.println(”WORD = “ + stok.sval); break;
    - case TT_NUMBER:
      - System.out.println(”NUM = “ + stok.nval); break;
    - case PLUS:
      - System.out.printlN(”PLUS”); break;
    - }
  - }
- }

Initialization

Configuration

Scanning

java.io.StreamTokenizer

Initialization: It's like using Java I/O!

- Constructor: StreamTokenizer(Reader r)

- java.io.Reader - class for reading bytes
  - FileReader - read bytes from a File
  - StringReader - read bytes from a String