2nd Edition



Using ANTLR Like a Professional

Learn From Years of Experience in Building Languages with Just One Course



Contents

Course overview	4
On completion of	3
The Professional Edition	4
The Standard Edition	7
The Price	9
ANTLR is the best	10
Testimonials	11
Experts in the field	12
Enterprise solutions	13

On completion of this course you will walk away with:

- A complete video course on parsing and ANTLR 4
- Understand the principles of parsing
- Learn to parse programming languages and data formats
- Create a parser quickly with ANTLR
- Get started on building compilers

The Professional Edition

The focus of the **Professional Edition** is to prepare a student to deal with all the typical issues they can face when building parsers for professional use. **This can be parser build for clients or that must be integrated in larger company projects.**

- we **reviewed and updated the included code** of all previous lessons. We updated all code to the latest coding standards and libraries. You will have the code in JavaScript, C# and Python. This required a bit of work, especially on the JavaScript and C# versions of the code
- we added **6 new lessons** on transforming a parse tree into an AST and how to deal with performance issues
- we added a Guide to best practices for ANTLR parsers.

Course Overview

- The **4 HR**+ video course
- The Guide to Best practices for ANTLR
- The whole code in Python, JavaScript and C# of

the projects and examples in the video course

- A 50+ pages manual with cheatsheets for ANTLR
- 5 Example Grammars explained and commented ranging from data formats to programming languages (JSON, DOT, HTML, SQLite and Kotlin)
- English subtitles and transcripts of all lessons in the video course
- You can download all the slides used in the video course
- Invitation to the reserved area of the Strumenta
 community on Language Engineering



Is this course right for you?

This course is for you if you are:

- a company looking for a professional internal training for employers professional development

- language engineer or architect who want to learn ANTLR and quickly apply your knowledge.

- professional developer who are looking to learn and start quickly to use ANTLR

- students, professors, researchers.

Table of content

- Introduction to Parsers. What are parsers and what they are good for.
 Explanation of the elements of a parser
- Introduction to ANTLR v4. Setting up ANTLR v4, the development environment and how to use the command line tools
- Grammar. The basic format of grammar and its elements. How to write Lexer and Parser rules and their typical patterns
- Listeners and Visitors. What are Listeners and Visitors and how you can use them to manage the results of the parsing
- Errors. Reporting errors and dealing with them
- **Testing.** How to test grammars, listeners and visitors
- Actions. The more advanced (but less clean) way to work the results of parsing
- Semantic Predicates. What are semantic predicates and how to use them
- How to Deal With Expressions. Why expressions are difficult to parse and two ways of dealing with them
- Lexical Modes. What are lexical modes and how you can use them to parse documents with multiple languages
- Tips and Tricks. A few tips and tricks to use ANTLR

- Parsing Real Programming Languages.
 What are and how to solve typical issues with parsing a real programming language
- **Designing a Grammar.** How to design a grammar, starting from scratch or from an existing one
- What Comes Next. A short introduction to other elements you need to build a compiler and how to integrate them with ANTLR
- (NEW) From Parse Tree to Abstract
 Syntax. Tree How to get from a parse tree to an Abstract Syntax Tree.
 Understanding the process and what things to transform
- (NEW) Creating an AST. Full example of transforming a parse tree in an AST
- (NEW) (Professional) Transforming
 Complex Parse Trees. Handling complex
 parse trees
- (NEW) (Professional) Setting Up for Performance. The mistakes to avoid in the organization of your ANTLR setup and grammar
- (NEW) (Professional) Changes to Improve Performance. The patterns to use and the ones to avoid in your grammar rules
- (NEW) (Professional) Tweaking ANTLR to Improve Performance. How to tweak ANTLR to squeeze more performance from the same grammar

More in details

The Professional Edition has 4 more lessons (bringing the total to 20 lessons) than the Standard edition. Therefore, the second version of the Professional Edition has 6 more lesson than the first edition. The first new three lessons explain how to transform a parse tree into an AST (Abstract Syntax Tree), while the other new three lessons deal with performance.

The first new lesson explains the issues that a parse tree has and how an AST can solve them. The second lesson shows a practical example in which we transform the typical elements of the parse tree into components of the AST. The third lesson shows how to handle complex parse trees that contain interdependent expressions.

The lessons about performance span the whole debate about performance issues. We talk about all the aspects of ANTLR that can affect performance and how to get the best performance for your parser. We provide guidelines to make your decisions. We list the mistakes to avoid in the organization of your ANTLR setup and grammar. We use our experience to provide the patterns to use and the ones to avoid in your grammar rules. We show how to tweak ANTLR to squeeze more performance from the same grammar.

Best practice ANTLR

In addition to these lessons for video course, the Professional Edition contains the Guide to Best Practices. This is an agile Ebook of 30 pages containing the principles we use when we create our parsers. These ranges from the ones we keep in mind when writing grammar rules to the ones we use to organize and design the whole grammar. In some sense this guide is the summary of the whole course: we organize all we have talked about in the course in a practical guide to keep close when you are creating a new parser.

The Standard Edition

The focus of the **Standard Edition** is to give to our students all the information they need to create effective and productive parsers that can be used for their projec

- we **reviewed and updated the included code** of all previous lessons. We updated all code to the latest coding standards and libraries. You will have the code in JavaScript, C# and Python. This required a bit of work, especially on the JavaScript and C# versions of the code
- we added 6 new lessons on transforming a parse tree into an AST and improving the performance of your parsers.

In the second edition we have reviewed the code and added 2 new lessons to show why and how to create an AST (Abstract Syntax Tree). The first new lesson explains the issues that a parse tree has and how an AST can solve them. The second lesson shows a practical example in which we transform the typical elements of the parse tree into components of the AST.

Course Overview

- The whole code in Python, JavaScript and C# of the projects and examples in the video course
- A 50+ pages manual with cheatsheets for ANTLR
- 5 Example Grammars explained and commented

ranging from data formats to programming

languages (JSON, DOT, HTML, SQLite and Kotlin)

- 16 video lessons for 3 and 1/2 HR+ video course
- English subtitles and transcripts of all lessons in the video course
- You can download all the slides used in the video course
- Invitation to the reserved area of the Strumenta

community on Language Engineering



Is this course right for you?

This course is for you if you are:

- an amateur looking to learn ANTLR

- if you want to create effective and productive parser

- you never use ANTLR and you want to start from the scratch

- students, professors, researchers.

Table of content

- Introduction to Parsers. What are parsers and what they are good for.
 Explanation of the elements of a parser
- Introduction to ANTLR v4. Setting up ANTLR v4, the development environment and how to use the command line tools
- Grammar. The basic format of grammar and its elements. How to write Lexer and Parser rules and their typical patterns
- Listeners and Visitors. What are Listeners and Visitors and how you can use them to manage the results of the parsing
- Errors. Reporting errors and dealing with them
- **Testing.** How to test grammars, listeners and visitors
- Actions. The more advanced (but less clean) way to work the results of parsing
- Semantic Predicates. What are semantic predicates and how to use them
- How to Deal With Expressions. Why expressions are difficult to parse and two ways of dealing with them
- Lexical Modes. What are lexical modes and how you can use them to parse documents with multiple languages
- Tips and Tricks. A few tips and tricks to use ANTLR

- Parsing Real Programming Languages.
 What are and how to solve typical issues with parsing a real programming language
- **Designing a Grammar.** How to design a grammar, starting from scratch or from an existing one
- What Comes Next. A short introduction to other elements you need to build a compiler and how to integrate them with ANTLR
- (NEW) From Parse Tree to Abstract
 Syntax. Tree How to get from a parse tree to an Abstract Syntax Tree.
 Understanding the process and what things to transform
- (NEW) Creating an AST. Full example of transforming a parse tree in an AST

Learn From Years of Experience in Building Languages with Just One Course

Parsing is more than reading files.

Parsing allows you to extract the data you want from a textual input in any format. It could be a text file, a binary file or source code.

You could parse an Nginx or Apache log file, a JPG file, a Java, Python or any computer language file.

Knowing how to parse means knowing how to analyze and understand any format.

You can build a parser from scratch, but using a tool to generate a parser is much easier and quicker.

This course **saves you time and give you the answers** you need to create and use parsers with ANTLR 4.

It explains the theory you need to work, it puts together all the **necessary information.**

It shows how to use ANTLR 4 and the tricks that save you time.

The Price

679\$ for the Professional Version

329\$ for the Standard Version

ANTLR is the best way to parse

ANTLR is the parser generator tool you want to use to build your parser.

At the forefront of research, but also production-ready and battle-tested.

Created by Terence Parr and developed in 1989. It is more than just a library.

ANTLR is a whole set of tools to help you be productive: from debugging grammars to visualize the result of parsing. **It hides the complications and the details.**

You do not need to know the theory behind parsing algorithms.

You do not need to know what left-recursive expressions are or what is context-sensitive parsing. No need to write unnaturally complicated rules.

You can build any grammar you need to be productive and ANTLR will make it work. **Use it in every parsing project.**

You can write just one grammar and let ANTLR generate parsers in many languages (Java, C#, Python, etc.)

Testimonials

66

I just bought 'How to create a pragmatic, lightweight language' book and I want to say thank you. I have 'Using ANTLR Like A Professional: Basic Edition' course too. I'm not up to day with any of these materials because I'm learning parsing techniques first, specially building recursive descent parsers by hand (just for getting the ability) but I'm pretty sure I'll be getting my hands dirty with these wonderful resources.

August 3, 2020 - Irwin Rodriguez

I just finished watching the 'Using ANTLR Like A Professional' video course, and I can now see how I can rewrite our existing parsers!

August 17, 2020 - Norman Jaffe

66

Just wanted to take the opportunity to say thanks. ANTLR is a BIG improvement over yacc/lex, and your support for it most commendable. Managed to get my tired old brain around it in a day. Nice work! Brad Cox, Inventor of Objective-C

Experts in the field

Absorb the experiences and passion of our professionals, save time and learn from real-life experience.



Federico Tomassetti

Founder and Language Architect at Strumenta

Federico has got his PhD in Language Engineering studying between Italy and Germany. In his role as a Software Architect he helps Clients and Partners to assess their situation and identify the best option to solve their problem. He is very passionate about Language Engineering and he contributes to the community by writing articles, participating in Program Committees of different conferences, giving presentations and interviewing other actors in this field.

Gabriele Tomassetti

Language Engineer at Strumenta

Gabriele started developing software to make a football management simulation and never stopped, although now he works on more useful software. He loves C# and Kotlin, but he can handle a few other languages.

Enterprise solutions

We can offer our Professional Edition course as an internal training for your team. Our enterprise training can also include coaching.

This can be the ideal solution to support your team in maintaining or developing ANTLR software.

Contact us directly to find out how enterprise licenses works.

If you are looking for a customized solution please contact us, we can find a tailored solution together.

Talk to us

If you want to know more or have questions about the course you can contact us at **info@strumenta.com**.

Do You Need ANTLR Consulting? Visit our website www.strumenta.com

Powered by

