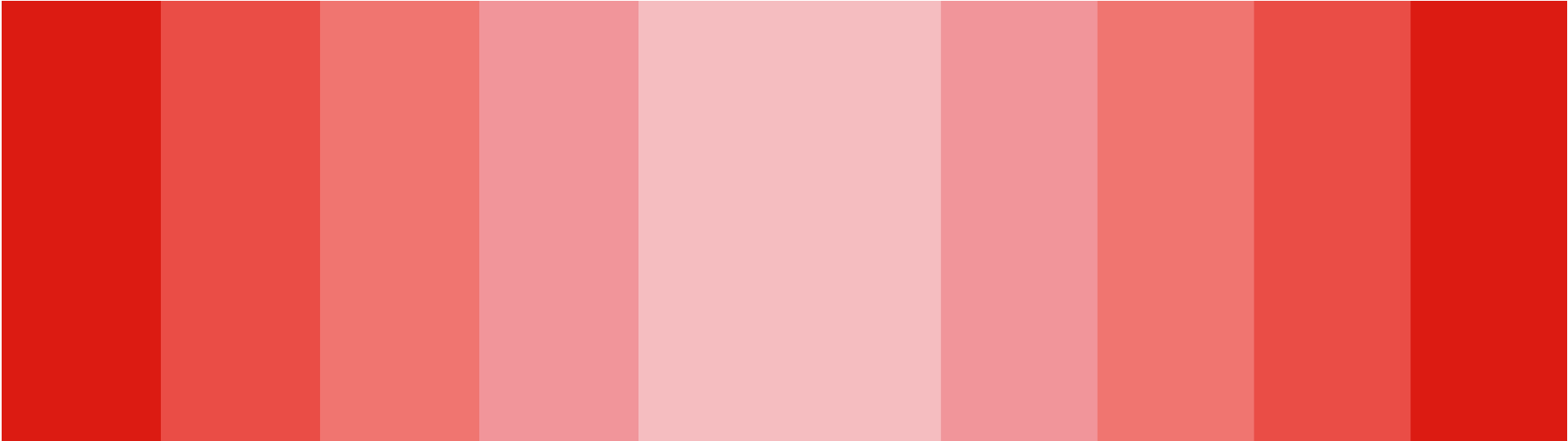




ANTLR



Agenda

1. Setting up the environment
2. Build Java files
3. Run

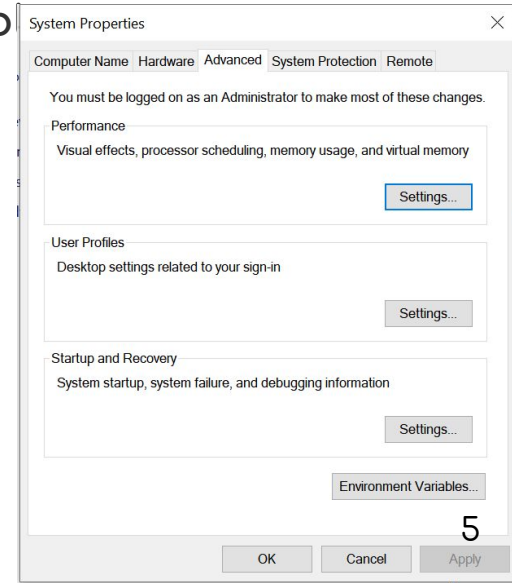
- ANTLR is really two things: a tool that translates your grammar to a parser/lexer in Java (or other target language) and the runtime needed by the generated parsers/lexers.
- Even if you are using the ANTLR **IntelliJ** plug-in or ANTLRWorks to run the ANTLR tool, the generated code will still need the runtime library.

Setup

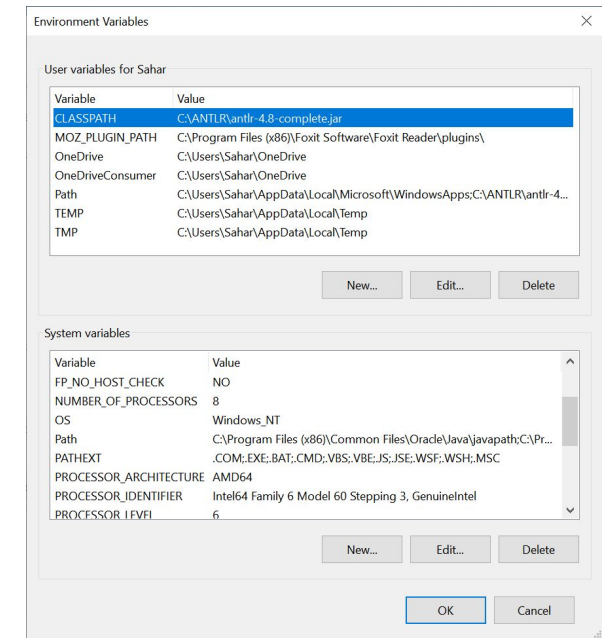
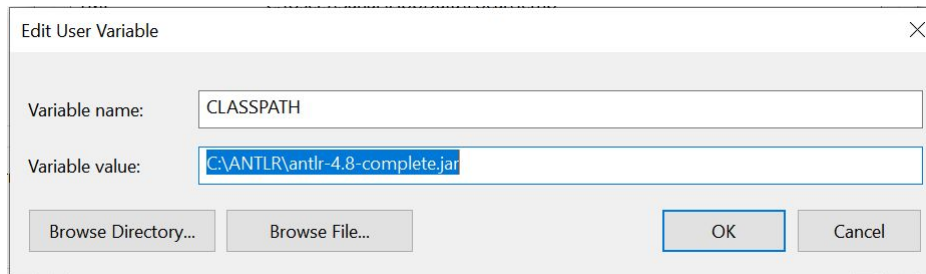
1. Install Java JDK (version 1.6 or higher)
2. Download antlr-4.8-complete.jar (or whatever version) from <https://www.antlr.org/download/>
3. Save to your directory for Java libraries, say C:\ANTLR

Add antlr-4.8-complete.jar to CLASSPATH

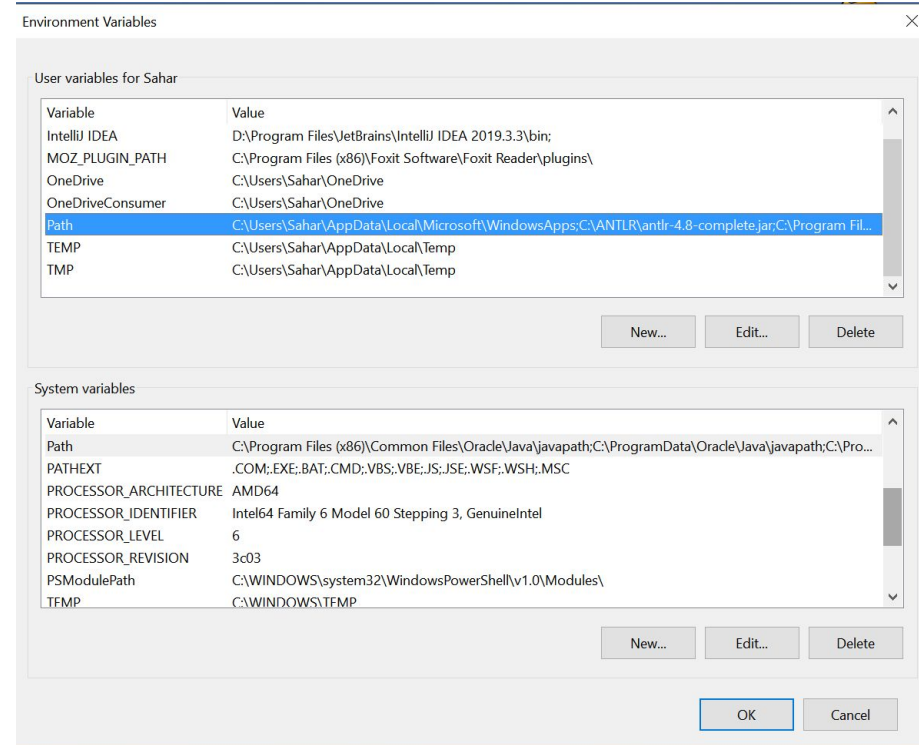
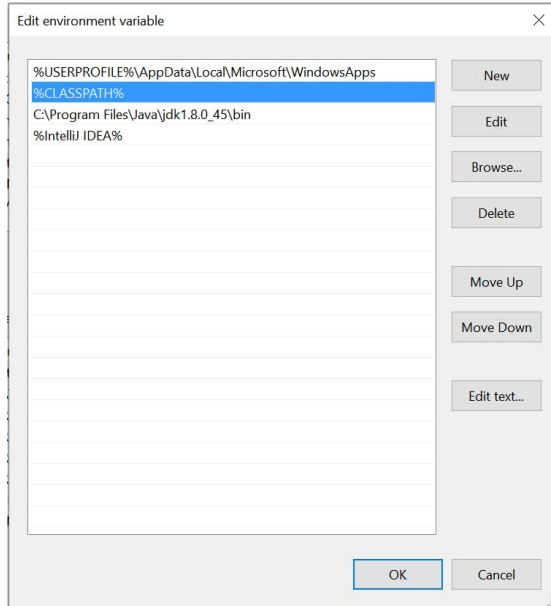
1. Open Up File Explorer
2. Right Click on This PC
3. Select Properties
4. On the left hand side of the new window select **Advanced system settings**
5. On the bottom of this screen select **Environment Variables**



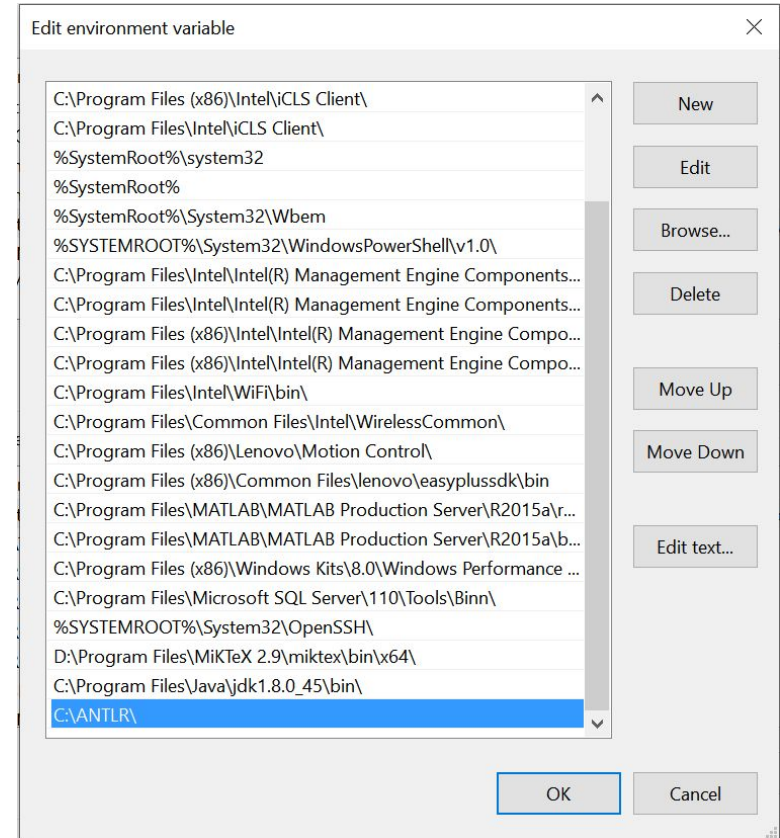
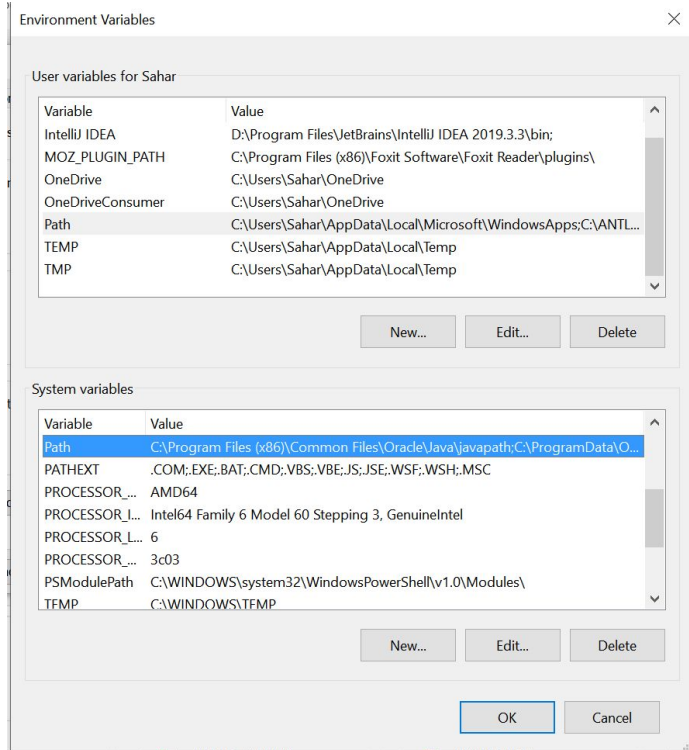
This is a new computer with no CLASSPATH variable so I had to add a new one. If you have used other Java Libraries it is possible that you may already have CLASSPATH defined. If you see CLASSPATH under System Variables append it with C:\ANTLR\antlr-4.8-complete.jar; If you do not have a CLASSPATH variable defined, select ****New...**** and then enter CLASSPATH for the Variable Name, and C:\ANTLR\antlr-4.8-complete.jar for the Variable value.



Add “%CLASSPATH%” to Path in User variables

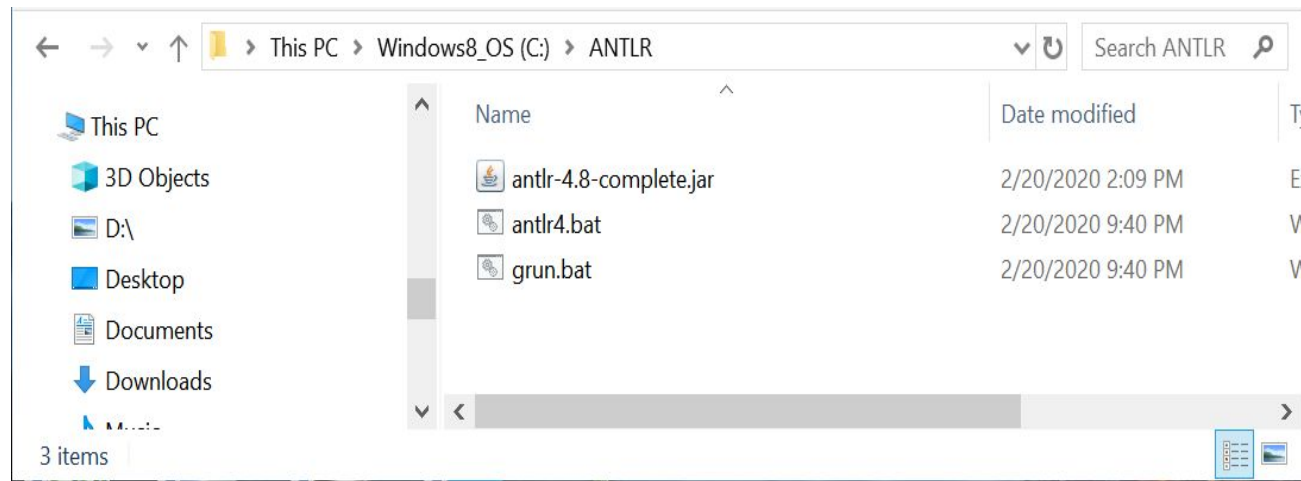


Add to System variables



Create two batch files in your ANTLR directory

1. antlr4.bat
2. grun.bat



antlr4.bat

1. Create a new file in C:\Javalib called antlr4.bat
(text document and change its “.txt” extension into “.bat”)
2. In this file enter `java org.antlr.v4.Tool %*`

```
java org.antlr.v4.Tool %*
```

3. Save this file

grun.bat

1. Create a new file in C:\Javalib called grun.bat
2. In this file enter

```
@ECHO OFF
```

```
SET TEST_CURRENT_DIR=%CLASSPATH:.,=%
```

```
if "%TEST_CURRENT_DIR%" == "%CLASSPATH%" ( SET CLASSPATH=.,%CLASSPATH% )
```

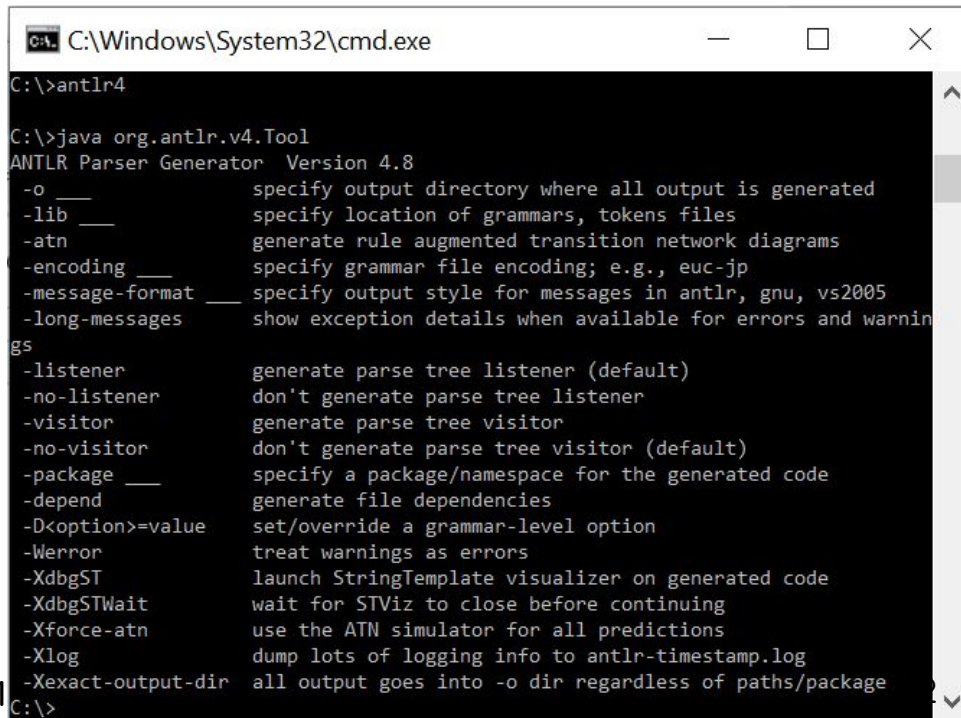
```
@ECHO ON
```

```
java org.antlr.v4.gui.TestRig %*
```

3. Save this file

Make Sure Everything Works

1. Open up a Command Prompt (you can type cmd in address bar of folder explorer)
2. Enter the command `antlr4`
3. You should see something like this:



```
C:\Windows\System32\cmd.exe
C:\>antlr4

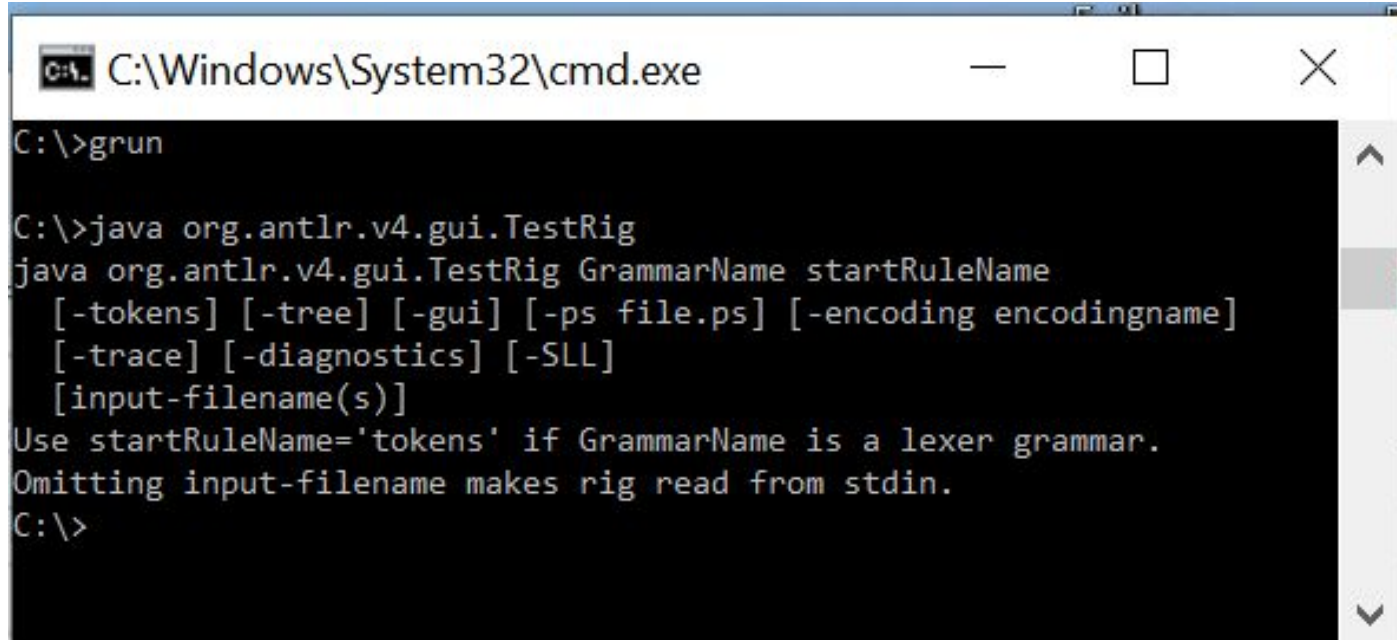
C:\>java org.antlr.v4.Tool
ANTLR Parser Generator Version 4.8

-o ____          specify output directory where all output is generated
-lib ____        specify location of grammars, tokens files
-atn             generate rule augmented transition network diagrams
-encoding ____   specify grammar file encoding; e.g., euc-jp
-message-format ____ specify output style for messages in antlr, gnu, vs2005
-long-messages   show exception details when available for errors and warnings

-listener        generate parse tree listener (default)
-no-listener     don't generate parse tree listener
-visitor         generate parse tree visitor
-no-visitor      don't generate parse tree visitor (default)
-package ____    specify a package/namespace for the generated code
-depend          generate file dependencies
-D<option>=value set/override a grammar-level option
-Werror          treat warnings as errors
-XdbgST          launch StringTemplate visualizer on generated code
-XdbgSTWait      wait for STViz to close before continuing
-Xforce-atn      use the ATN simulator for all predictions
-Xlog            dump lots of logging info to antlr-timestamp.log
-Xexact-output-dir all output goes into -o dir regardless of paths/package

C:\>
```

Enter the command grun



```
C:\Windows\System32\cmd.exe

C:\>grun

C:\>java org.antlr.v4.gui.TestRig
java org.antlr.v4.gui.TestRig GrammarName startRuleName
  [-tokens] [-tree] [-gui] [-ps file.ps] [-encoding encodingname]
  [-trace] [-diagnostics] [-SLL]
  [input-filename(s)]
Use startRuleName='tokens' if GrammarName is a lexer grammar.
Omitting input-filename makes rig read from stdin.
C:\>
```

Executing the instructions on Windows

1. define a lexer and parser grammar
2. invoke ANTLR: it will generate a lexer and a parser in your target language (e.g., Java, Python, C#, Javascript)
3. use the generated lexer and parser: you invoke them passing the code to recognize and they return to you an AST

Example 1

1. Create a new file “Hello.g4” in your code folder
2. Add your grammar in it and save it

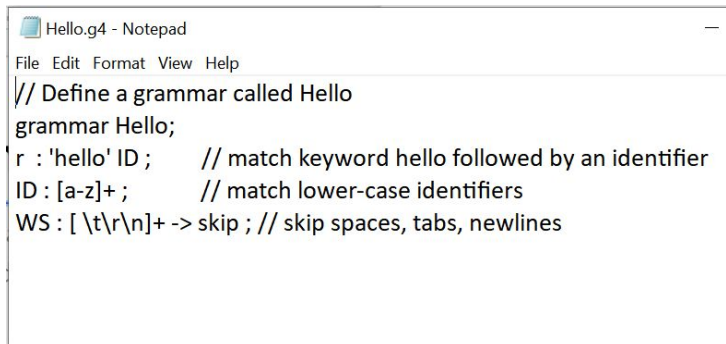
```
// Define a grammar called Hello
```

```
grammar Hello;
```

```
r : 'hello' ID ;           // match keyword hello followed by an identifier
```

```
ID : [a-z]+ ;             // match lower-case identifiers
```

```
WS : [ \t\r\n]+ -> skip ; // skip spaces, tabs, newlines
```



```
Hello.g4 - Notepad
File Edit Format View Help
// Define a grammar called Hello
grammar Hello;
r : 'hello' ID ;           // match keyword hello followed by an identifier
ID : [a-z]+ ;             // match lower-case identifiers
WS : [ \t\r\n]+ -> skip ; // skip spaces, tabs, newlines
```

N.B. File name must be the same as your grammar (Case sensitive)

Example 1: run ANTLR the tool on it

Open command line and change directory to the folder containing your file (or just type cmd in the address bar of your folder)

\$ cd d:\ANTLRCode	//change directory
\$ antlr4 Hello.g4	//Run antlr on your grammar
\$ javac Hello*.java	//Compile your java class file



```
D:\>cd d:\ANTLRCode

d:\ANTLRCode>antlr4 Hello.g4

d:\ANTLRCode>java org.antlr.v4.Tool Hello.g4
d:\ANTLRCode>javac Hello*.java

d:\ANTLRCode>
```


Example 1: Test

```
$ grun Hello r -tokens
```

```
// r is the start of your regular expression in the Hello file
```

```
// -tokens option tokenize the input string.
```

hello world //input your string then

press Ctrl+Z //(^Z) EOF in windows or (^D) in unix

Output

grun Hello r -tokens

```
Microsoft Windows [Version 10.0.18362.657]
(c) 2019 Microsoft Corporation. All rights reserved.

D:\ANTLRCode\Hello>$ grun Hello r -tokens
'$' is not recognized as an internal or external command,
operable program or batch file.

D:\ANTLRCode\Hello>grun Hello r -tokens

D:\ANTLRCode\Hello>java org.antlr.v4.gui.TestRig Hello r -tokens
hello csen
^Z
[@0,0:4='hello',<'hello'>,1:0]
[@1,6:9='csen',<ID>,1:6]
[@2,12:11='<EOF>','<EOF>',2:0]
D:\ANTLRCode\Hello>
```

Tokens

Actions

- Lexer actions can appear anywhere, not just at the end of the outermost alternative.
- The lexer executes the actions at the appropriate input position, according to the placement of the action within the rule.
- To execute a single action for a rule that has multiple alternatives, you can enclose the alts in parentheses and put the action afterwards:
- `END : ('endif'|'end') {System.out.println("found an end");} ;`

- The action conforms to the syntax of the target language. ANTLR copies the action's contents into the generated code verbatim; there is no translation of expressions like `$x.y` as there is in parser actions.
- Only actions within the **outermost token** rule are executed. In other words, if `STRING` calls `ESC_CHAR` and `ESC_CHAR` has an action, that action is not executed when the lexer starts matching in `STRING`.

Example 2

grammar Expr;

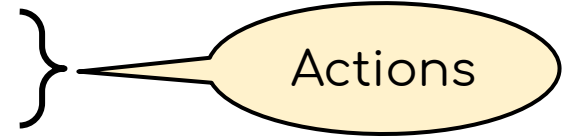
prog: (expr NEWLINE)* ;

expr: expr ('*'|'/') expr {System.out.println("111");}
 | expr ('+'|'-') expr {System.out.println("222");}
 | INT
 | '(' expr ')'

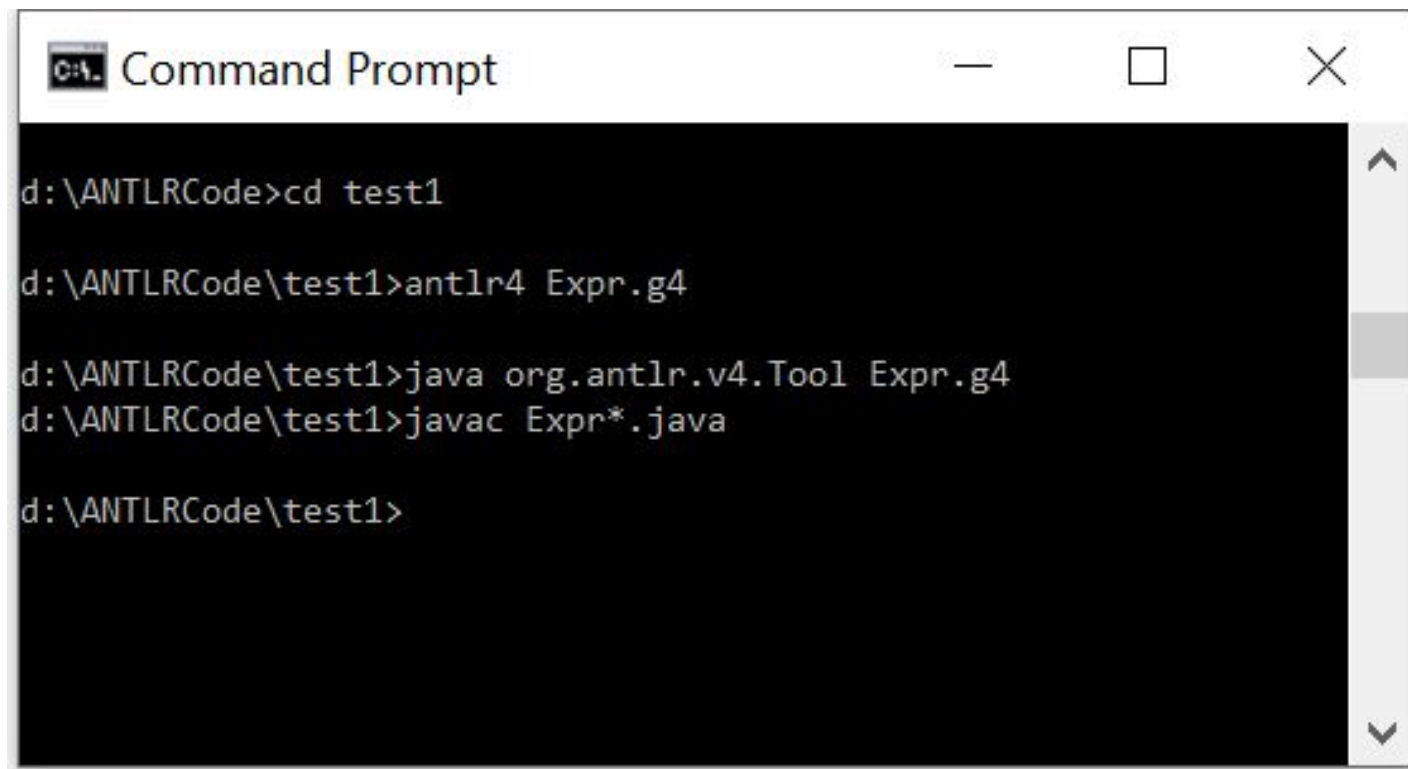
;

NEWLINE : [\r\n]+ ;

INT : [0-9]+ ;



Example 2



```
d:\ANTLRCode>cd test1

d:\ANTLRCode\test1>antlr4 Expr.g4

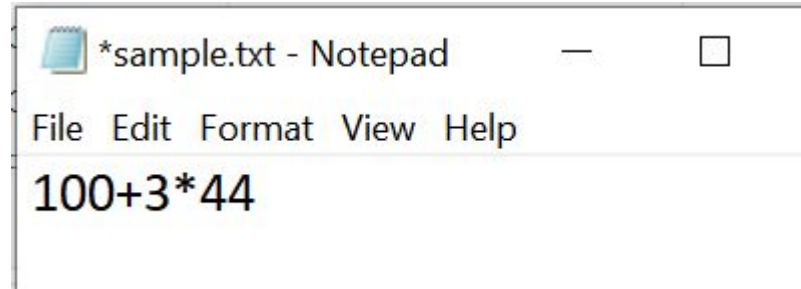
d:\ANTLRCode\test1>java org.antlr.v4.Tool Expr.g4
d:\ANTLRCode\test1>javac Expr*.java

d:\ANTLRCode\test1>
```

Example 2: Test

Create a text file that contains your string

$100+3*44$



Example 2: Test

Start of
regular
expression

\$ grun Expr **prog** -tokens sample.txt

```
D:\ANTLRCode\test1>java org.antlr.v4.gui.TestRig Expr prog -tokens sample.txt
[@0,0:2='100',<INT>,1:0]
[@1,3:3='+',<'+'>,1:3]
[@2,4:4='3',<INT>,1:4]
[@3,5:5='*',<'*'>,1:5]
[@4,6:7='44',<INT>,1:6]
[@5,8:11='\r\n\r\n',<NEWLINE>,1:8]
[@6,12:11='<EOF>',<EOF>,3:0]
111
222
D:\ANTLRCode\test1>
```

Tokens

Actions

References

<https://www.youtube.com/watch?v=O0D0Lc5syGA>

<https://www.antlr.org/>

<https://levlaz.org/setting-up-antlr4-on-windows/>

<https://tomassetti.me/antlr-mega-tutorial/>