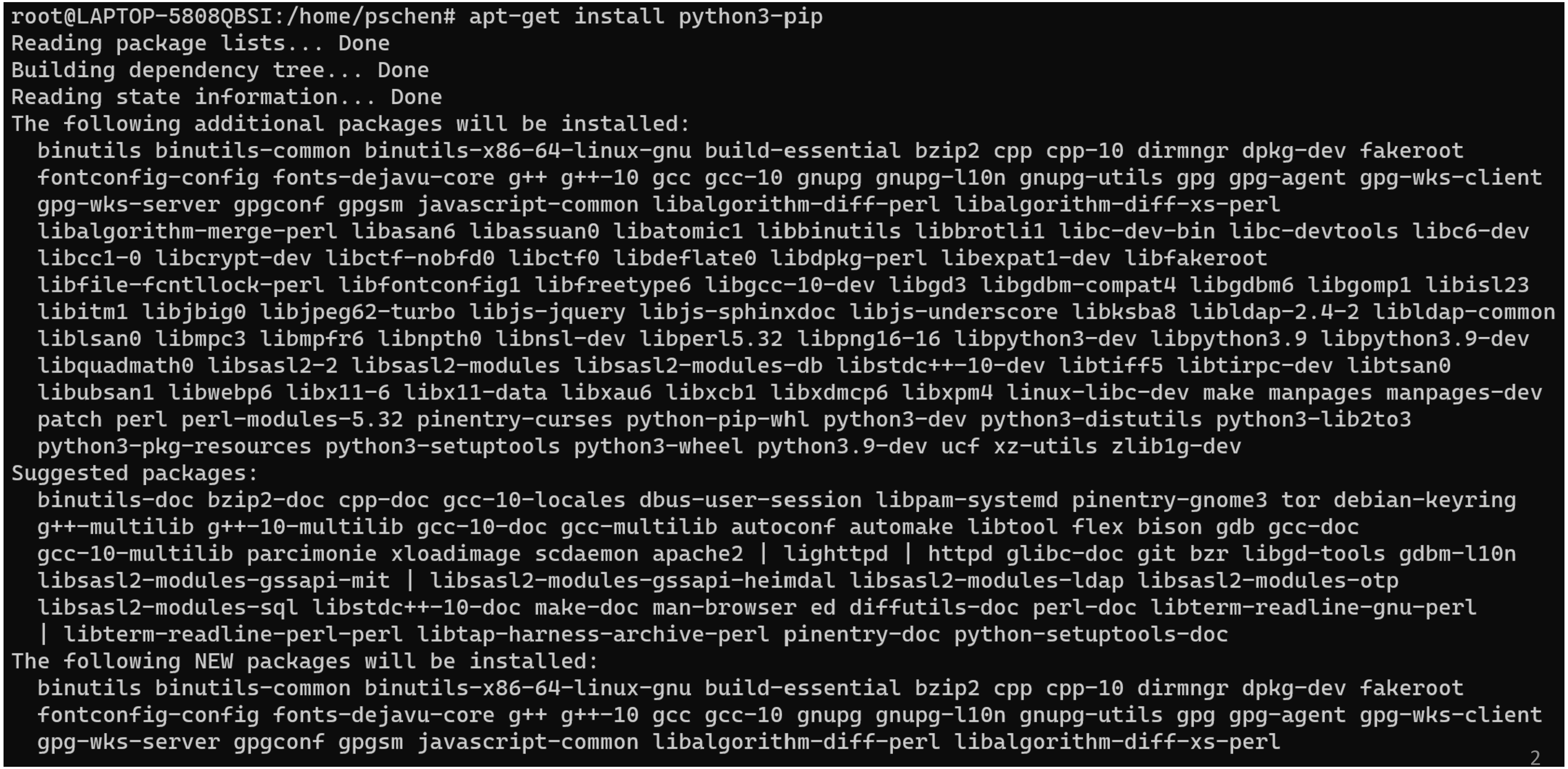
ANTLR v4

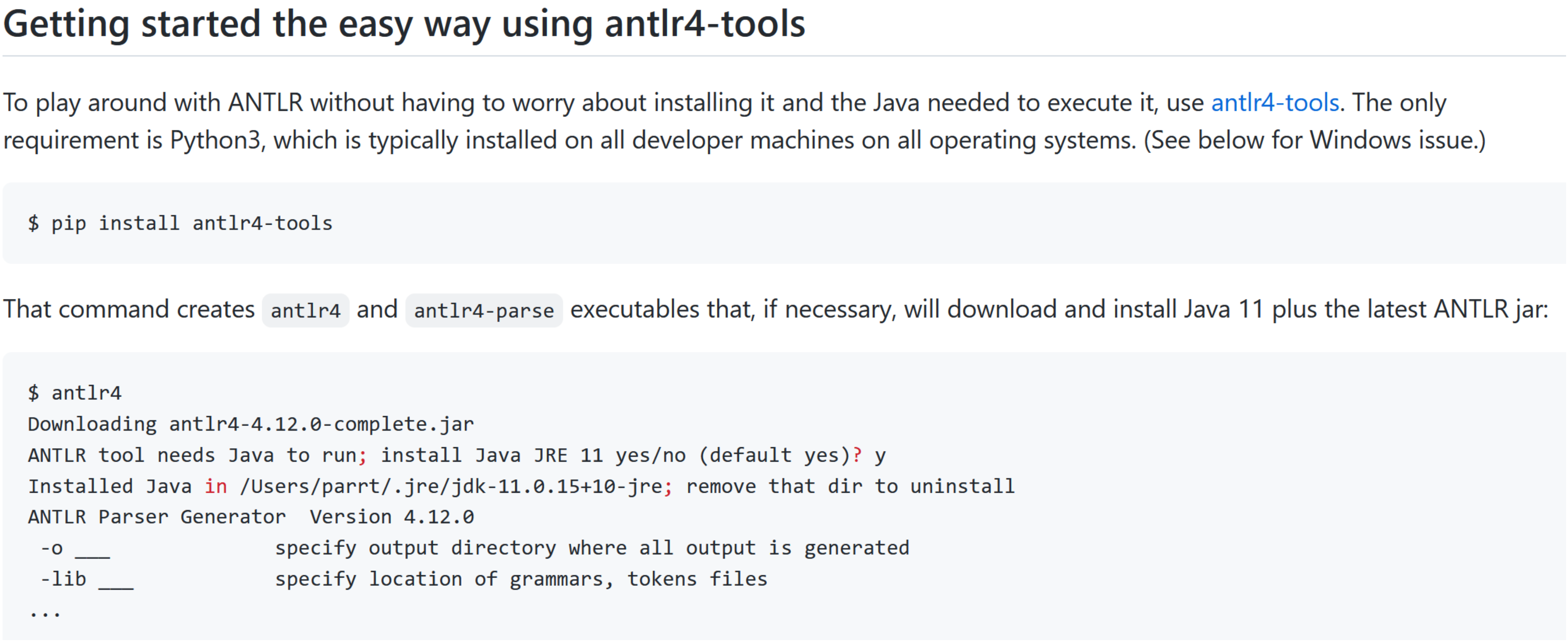
Installation: ANTLR4 (1)

- **`$ apt-get install python3 python3-pip`**



```
root@LAPTOP-5808QBSI:/home/pschen# apt-get install python3-pip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential bzip2 cpp cpp-10 dirmngr dpkg-dev fakeroot
  fontconfig-config fonts-dejavu-core g++ g++-10 gcc gcc-10 gnupg gnupg-l10n gnupg-utils gpg gpg-agent gpg-wks-client
  gpg-wks-server gpgconf gpgsm javascript-common libalgorithm-diff-perl libalgorithm-diff-xs-perl
  libalgorithm-merge-perl libasan6 libassuan0 libatomic1 libbinutils libbrotli1 libc-dev-bin libc-devtools libc6-dev
  libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libdeflate0 libdpkg-perl libexpat1-dev libfakeroot
  libfile-fcntllock-perl libfontconfig1 libfreetype6 libgcc-10-dev libgd3 libgdbm-compat4 libgdbm6 libgomp1 libisl23
  libitm1 libjbig0 libjpeg62-turbo libjs-jquery libjs-sphinxdoc libjs-underscore libksba8 libldap-2.4-2 libldap-common
  liblsan0 libmpc3 libmpfr6 libnpth0 libnsl-dev libperl5.32 libpng16-16 libpython3-dev libpython3.9 libpython3.9-dev
  libquadmath0 libsasl2-2 libsasl2-modules libsasl2-modules-db libstdc++-10-dev libtiff5 libtirpc-dev libtsan0
  libubsan1 libwebp6 libx11-6 libx11-data libxau6 libxcb1 libxdmcp6 libxpm4 linux-libc-dev make manpages manpages-dev
  patch perl perl-modules-5.32 pinentry-curses python-pip-whl python3-dev python3-distutils python3-lib2to3
  python3-pkg-resources python3-setuptools python3-wheel python3.9-dev ucf xz-utils zlib1g-dev
Suggested packages:
  binutils-doc bzip2-doc cpp-doc gcc-10-locales dbus-user-session libpam-systemd pinentry-gnome3 tor debian-keyring
  g++-multilib g++-10-multilib gcc-10-doc gcc-multilib autoconf automake libtool flex bison gdb gcc-doc
  gcc-10-multilib parcimonie xloadimage scdaemon apache2 | lighttpd | httpd glibc-doc git bzr libgd-tools gdbm-l10n
  libsasl2-modules-gssapi-mit | libsasl2-modules-gssapi-heimdal libsasl2-modules-ldap libsasl2-modules-otp
  libsasl2-modules-sql libstdc++-10-doc make-doc man-browser ed diffutils-doc perl-doc libterm-readline-gnu-perl
  | libterm-readline-perl-perl libtap-harness-archive-perl pinentry-doc python-setuptools-doc
The following NEW packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential bzip2 cpp cpp-10 dirmngr dpkg-dev fakeroot
  fontconfig-config fonts-dejavu-core g++ g++-10 gcc gcc-10 gnupg gnupg-l10n gnupg-utils gpg gpg-agent gpg-wks-client
  gpg-wks-server gpgconf gpgsm javascript-common libalgorithm-diff-perl libalgorithm-diff-xs-perl
```

Installation: ANTLR4 (2)



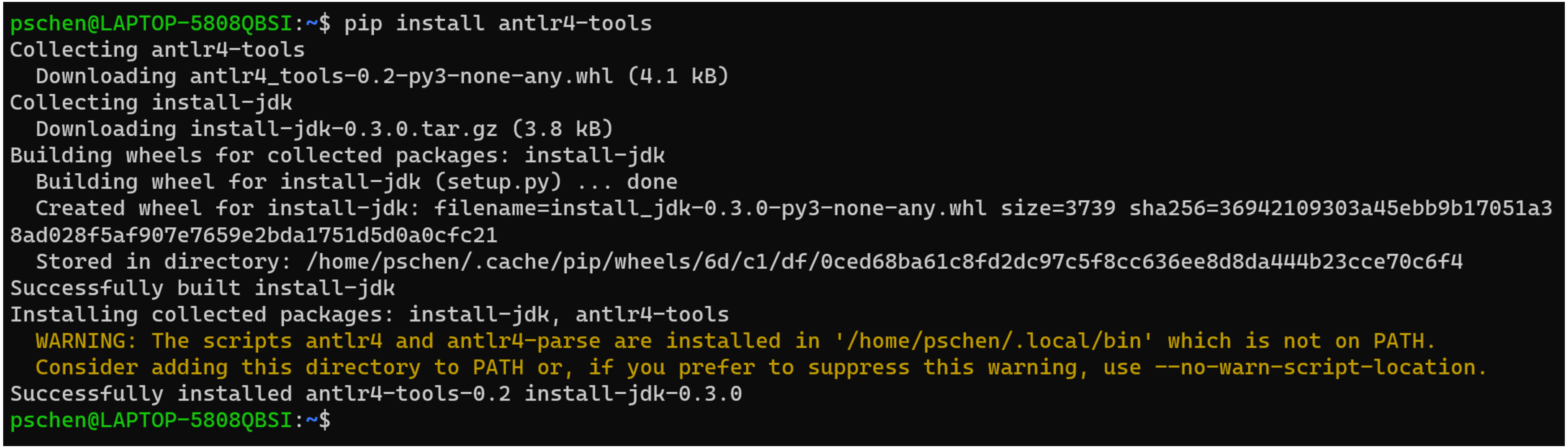
Getting started the easy way using antlr4-tools

To play around with ANTLR without having to worry about installing it and the Java needed to execute it, use antlr4-tools. The only requirement is Python3, which is typically installed on all developer machines on all operating systems. (See below for Windows issue.)

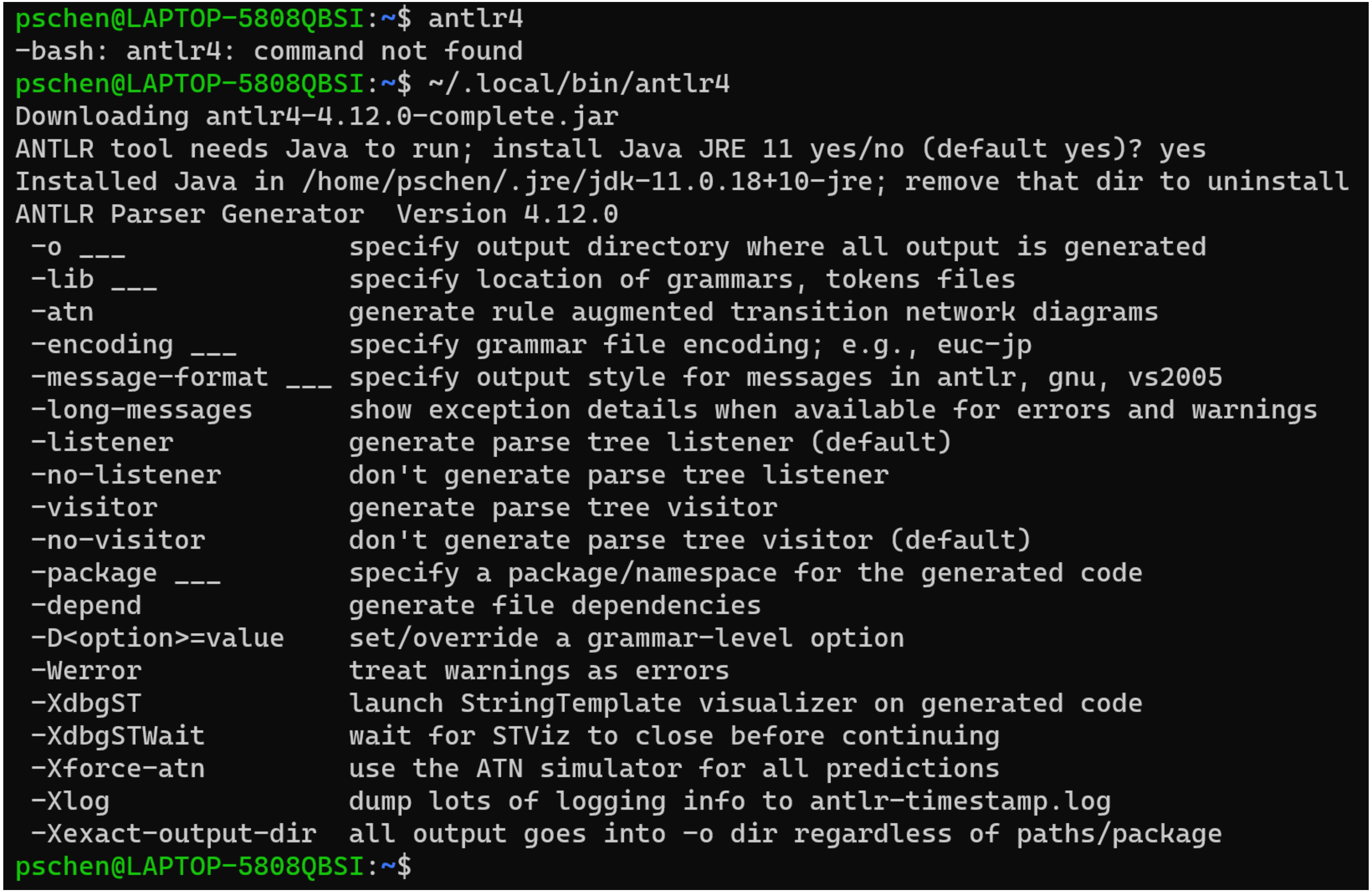
```
$ pip install antlr4-tools
```

That command creates `antlr4` and `antlr4-parse` executables that, if necessary, will download and install Java 11 plus the latest ANTLR jar:

```
$ antlr4
Downloading antlr4-4.12.0-complete.jar
ANTLR tool needs Java to run; install Java JRE 11 yes/no (default yes)? y
Installed Java in /Users/parrt/.jre/jdk-11.0.15+10-jre; remove that dir to uninstall
ANTLR Parser Generator  Version 4.12.0
 -o ___              specify output directory where all output is generated
 -lib ___            specify location of grammars, tokens files
...
```



```
pschen@LAPTOP-5808QBSI:~$ pip install antlr4-tools
Collecting antlr4-tools
  Downloading antlr4_tools-0.2-py3-none-any.whl (4.1 kB)
Collecting install-jdk
  Downloading install-jdk-0.3.0.tar.gz (3.8 kB)
Building wheels for collected packages: install-jdk
  Building wheel for install-jdk (setup.py) ... done
  Created wheel for install-jdk: filename=install_jdk-0.3.0-py3-none-any.whl size=3739 sha256=36942109303a45ebb9b17051a3
8ad028f5af907e7659e2bda1751d5d0a0cfc21
  Stored in directory: /home/pschen/.cache/pip/wheels/6d/c1/df/0ced68ba61c8fd2dc97c5f8cc636ee8d8da444b23cce70c6f4
Successfully built install-jdk
Installing collected packages: install-jdk, antlr4-tools
  WARNING: The scripts antlr4 and antlr4-parse are installed in '/home/pschen/.local/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed antlr4-tools-0.2 install-jdk-0.3.0
pschen@LAPTOP-5808QBSI:~$
```



```
pschen@LAPTOP-5808QBSI:~$ antlr4
-bash: antlr4: command not found
pschen@LAPTOP-5808QBSI:~$ ~/.local/bin/antlr4
Downloading antlr4-4.12.0-complete.jar
ANTLR tool needs Java to run; install Java JRE 11 yes/no (default yes)? yes
Installed Java in /home/pschen/.jre/jdk-11.0.18+10-jre; remove that dir to uninstall
ANTLR Parser Generator  Version 4.12.0
 -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
pschen@LAPTOP-5808QBSI:~$
```

Installation: ANTLR4 (3)

```
$export PATH=/home/pschen/.local/bin:$PATH
```

- **`antlr4-4.12.0-complete.jar`**
  - /home/pschen/.m2/repository/org/antlr/antlr4/4.12.0/antlr4-4.12.0-complete.jar

```
$curl -O https://www.antlr.org/download/antlr-4.12.0-complete.jar
```

Wildcard Operator and Nongreedy Subrules

- Input: **/* this is a book. */**
- COMMENT : '/*’ (.)* '*/’;

- R.E. (…)?, (…)*, and (…)+ are greedy. They consume as much input as possible.
- ANTLR3
  - COMMENT : '/*’ (options{greedy=false;}:.)* '*/’;
- ANTLR4
  - COMMENT : '/*' (.)*? '*/';

```
import org.antlr.v4.runtime.CharStream;
import org.antlr.v4.runtime.Token;
import org.antlr.v4.runtime.CharStreams;

public class testLexer {
   public static void main(String[] args) {
      //CharStream input = new ANTLRFileStream(args[0]);
      CharStream input = CharStreams.fromFileName(args[0]);
      test1 lexer = new test1(input);
      Token token = lexer.nextToken();
      while (token.getType() != -1) {  // -1 is EOF.
        System.out.println("Token: "+ token.getType() + " “
                                             + token.getText());
        token = lexer.nextToken();
   }
}
```