```java
// Generated from c:/Users/Jorge
Melo/Desktop/python2asm_ProgSistBase1/grammar/PythonSubset.g4 by ANTLR 4.13.1
import org.antlr.v4.runtime.Lexer;
import org.antlr.v4.runtime.CharStream;
import org.antlr.v4.runtime.Token;
import org.antlr.v4.runtime.TokenStream;
import org.antlr.v4.runtime.*;
import org.antlr.v4.runtime.atn.*;
import org.antlr.v4.runtime.dfa.DFA;
import org.antlr.v4.runtime.misc.*;

@SuppressWarnings({"all", "warnings", "unchecked", "unused", "cast",
"CheckReturnValue", "this-escape"})
public class PythonSubsetLexer extends Lexer {
 static { RuntimeMetaData.checkVersion("4.13.1", RuntimeMetaData.VERSION); }

 protected static final DFA[] _decisionToDFA;
 protected static final PredictionContextCache _sharedContextCache =
  new PredictionContextCache();
 public static final int
  T__0=1, T__1=2, T__2=3, T__3=4, T__4=5, T__5=6, T__6=7, T__7=8, T__8=9,
  T__9=10, T__10=11, T__11=12, T__12=13, T__13=14, T__14=15, T__15=16, T__16=17,
  T__17=18, FOR=19, WHILE=20, IF=21, ELIF=22, ELSE=23, IN=24, AND=25, OR=26,
  NOT=27, TRUE=28, FALSE=29, INDENT=30, DEDENT=31, IDENTIFIER=32, INT=33,
  STRING=34, NEWLINE=35, WS=36, COMMENT=37;
 public static String[] channelNames = {
  "DEFAULT_TOKEN_CHANNEL", "HIDDEN"
 };

 public static String[] modeNames = {
  "DEFAULT_MODE"
 };

 private static String[] makeRuleNames() {
  return new String[] {
   "T__0", "T__1", "T__2", "T__3", "T__4", "T__5", "T__6", "T__7", "T__8",
   "T__9", "T__10", "T__11", "T__12", "T__13", "T__14", "T__15", "T__16",
   "T__17", "FOR", "WHILE", "IF", "ELIF", "ELSE", "IN", "AND", "OR", "NOT",
   "TRUE", "FALSE", "INDENT", "DEDENT", "IDENTIFIER", "INT", "STRING", "NEWLINE",
   "WS", "COMMENT"
  };
 }
 public static final String[] ruleNames = makeRuleNames();

 private static String[] makeLiteralNames() {
  return new String[] {
   null, "'='", "':'", "'range'", "'('", "')'", "','", "'=='", "'!='", "'>='",
   "'<='", "'>'", "'<'", "'+'", "'-'", "'*'", "'/'", "'%'", "'**'", "'for'",
   "'while'", "'if'", "'elif'", "'else'", "'in'", "'and'", "'or'", "'not'",
   "'True'", "'False'", "'INDENT'", "'DEDENT'"
  };
 }
```

```
  private static final String[] _LITERAL_NAMES = makeLiteralNames();
  private static String[] makeSymbolicNames() {
   return new String[] {
    null, null, null, null, null, null, null, null, null, null, null, null,
    null, null, null, null, null, null, null, "FOR", "WHILE", "IF", "ELIF",
    "ELSE", "IN", "AND", "OR", "NOT", "TRUE", "FALSE", "INDENT", "DEDENT",
    "IDENTIFIER", "INT", "STRING", "NEWLINE", "WS", "COMMENT"
   };
  }
  private static final String[] _SYMBOLIC_NAMES = makeSymbolicNames();
  public static final Vocabulary VOCABULARY = new VocabularyImpl(_LITERAL_NAMES,
_SYMBOLIC_NAMES);

  /**
   * @deprecated Use {@link #VOCABULARY} instead.
   */
  @Deprecated
  public static final String[] tokenNames;
  static {
   tokenNames = new String[_SYMBOLIC_NAMES.length];
   for (int i = 0; i < tokenNames.length; i++) {
    tokenNames[i] = VOCABULARY.getLiteralName(i);
    if (tokenNames[i] == null) {
     tokenNames[i] = VOCABULARY.getSymbolicName(i);
    }

    if (tokenNames[i] == null) {
     tokenNames[i] = "<INVALID>";
    }
   }
  }
```