# Tutorial
# Lexical Analysis

## Question 1.
Use ANTLR to write regular expressions describing a Pascal **identifier** that must begin with a lowercase letter ('a' to 'z'), but may continue with many characters which are lowercase letter or digit ('0' to '9'). [a-z][a-z0-9]*

## Question 2. đặt tên cho Regex, như đặt tên biến
A *regular definition* is used to name a regular expression and then the name is used in another regular expression. For example, given the following regular definition:

| | |
|---|---|
| letter | [a-z] |
| manyletter | letter+ |

fragment Letter: [a-z]
fragment Number: [0-9]
identifier: Letter(Letter | Number)*

In ANTLR, to define a *regular definition*, we use **fragment** as the following example:

fragment Letter: [a-z]; dùng keyword fragment để khai báo Regex
Manyletter Letter+ ; dùng Regex thông qua tên đã khai báo

Use *fragment* in ANTLR to rewrite the regular expression for the above token Identifier

## Question 3.
Use ANTLR to write regular expressions describing the following Pascal tokens:

a) For a number to be taken as "**real**" (or "floating point") format, it must either have a decimal point, or use scientific notation. For example, 1.0, 1e-12, 1.0e-12, 0.000000001 are all valid reals. At least one digit must exist on either side of a decimal point.

real = [+-]?\d+((\.\d+e[-+]?)|(\.|(e[-+]?)))\d+

b) **Strings** are made up of a sequence of characters between single quotes: 'string'. The single quote itself can appear as two single quotes back to back in a string: 'isn"t'.

string = ^'([^']|'')*'$

## Question 4.
Find regular expressions and state diagrams of the equivalent NFA for each of the following descriptions.

a) $\{a^n b^m | n \geq 0, m > 2\}$  a*b{3,} a*bb

b) $\{a^n b^m | n + m > 0, n + m \ is \ even\}$  ((aa)*|(bb)*|(a(aa)*b(bb)*)|((aa)+(bb)+))

c) $\{a^n b | n \ mod \ 3 \ = \ 1\}$  a(aaa)