**Análise Léxica – Python**

**Francisco Targa e Eduardo de Toledo**

Regra 1: STRING\_LITERAL

A regra léxica STRING\_LITERAL é responsável por reconhecer literais de strings no código Python e separá-los como tokens individuais para análise posterior pelo analisador sintático (parser).

**STRING\_LITERAL: '\'' (ESC | ~('\'' | '\\'))\* '\''**

**| '"' (ESC | ~('"' | '\\'))\* '"';**

**Entradas aceitas:**

String delimitada por aspas simples (''):

1. Entrada:



Nesse caso, a string "Hello, world!" é delimitada por aspas simples, e não contém caracteres de escape.

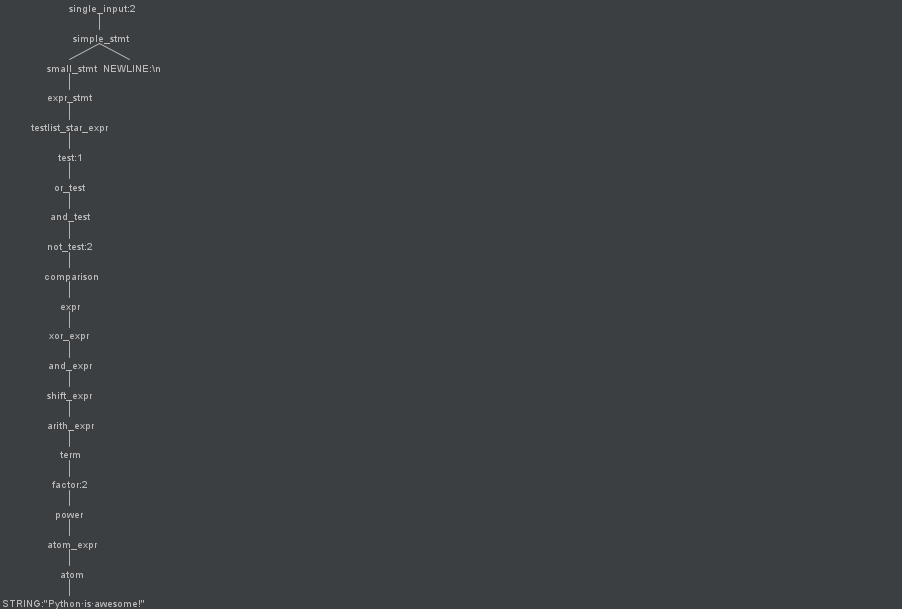


String delimitada por aspas duplas (""):

1. Entrada:



Nesse caso, a string "Python is awesome!" é delimitada por aspas duplas. Na linguagem Python, também é uma forma reconhecida de string.



**Entradas rejeitadas:**

String com apenas uma aspa:

Entrada:



Nesse caso, a string é iniciada com uma aspa simples, mas não é fechada com outra aspa simples. Como resultado, a regra STRING\_LITERAL não conseguirá reconhecer essa string, pois não segue o formato de abertura e fechamento esperado.

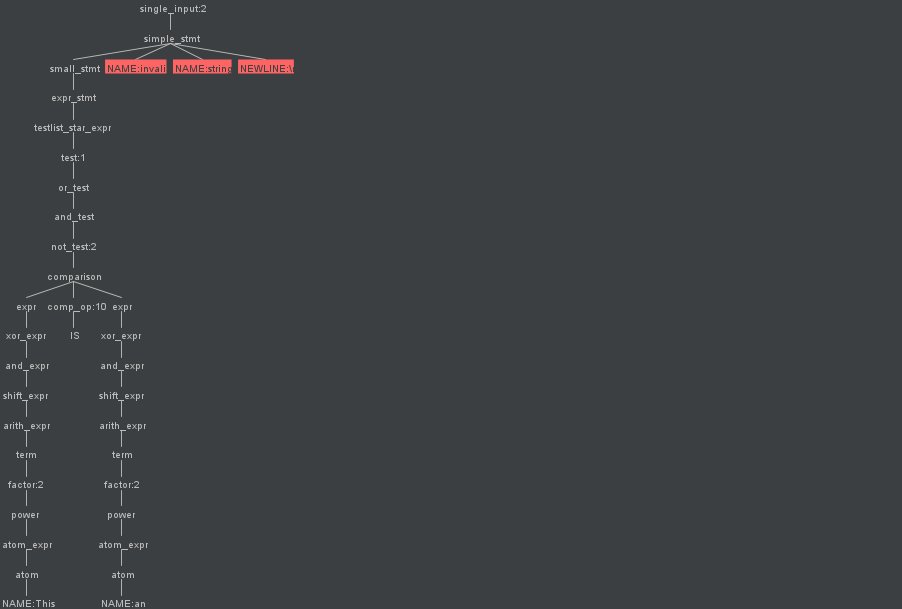


String não fechada:

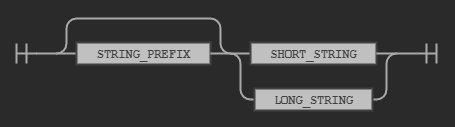
Entrada:



Nesse caso, a entrada não contém aspas no início e no final, portanto não é uma string literal válida de acordo com a regra STRING\_LITERAL.



**Diagrama de Sintaxe Equivalente (Railroad)**

****

Regra 2: IMPORT

A regra IMPORT é responsável por reconhecer declarações de importação em código Python. Essas declarações são usadas para importar módulos ou partes específicas de módulos em um programa Python.

**IMPORT: 'import' WS+ IMPORT\_NAME (WS\* ',' WS\* IMPORT\_NAME)\* WS\* (NEWLINE | ';')?;**

**Entradas aceitas:**

1. Importação de Módulo Simples:



Neste caso, a declaração de importação simples import math seria reconhecida pela regra IMPORT.



1. Importação de Múltiplos Itens



Aqui, a declaração from datetime import datetime, timedelta importa os itens datetime e timedelta do módulo datetime. Essa declaração também seria reconhecida pela regra IMPORT.



**Entradas rejeitadas:**

1. Importação Mal formatada



Neste exemplo, a declaração de importação está correta, mas a regra IMPORT espera que a declaração termine com uma nova linha ou um ponto e vírgula. Como não há nenhum desses terminadores na entrada, a regra IMPORT rejeitaria essa entrada.

1. Importação sem Nome de Módulo



Aqui, a declaração de importação não inclui o nome do módulo que está sendo importado. A regra IMPORT espera que após a palavra-chave import haja um nome de módulo válido, mas como isso está faltando, a entrada seria rejeitada.



**Diagrama de Sintaxe Equivalente (Railroad)**

****

Regra 3: COMMENT

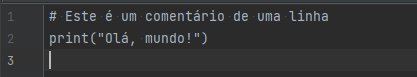
A regra COMMENT é responsável por reconhecer comentários no código Python. Comentários são partes ignoradas pelo interpretador/compilador.

**COMMENT : '#' ~[\r\n\f]\* -> channel(HIDDEN);**

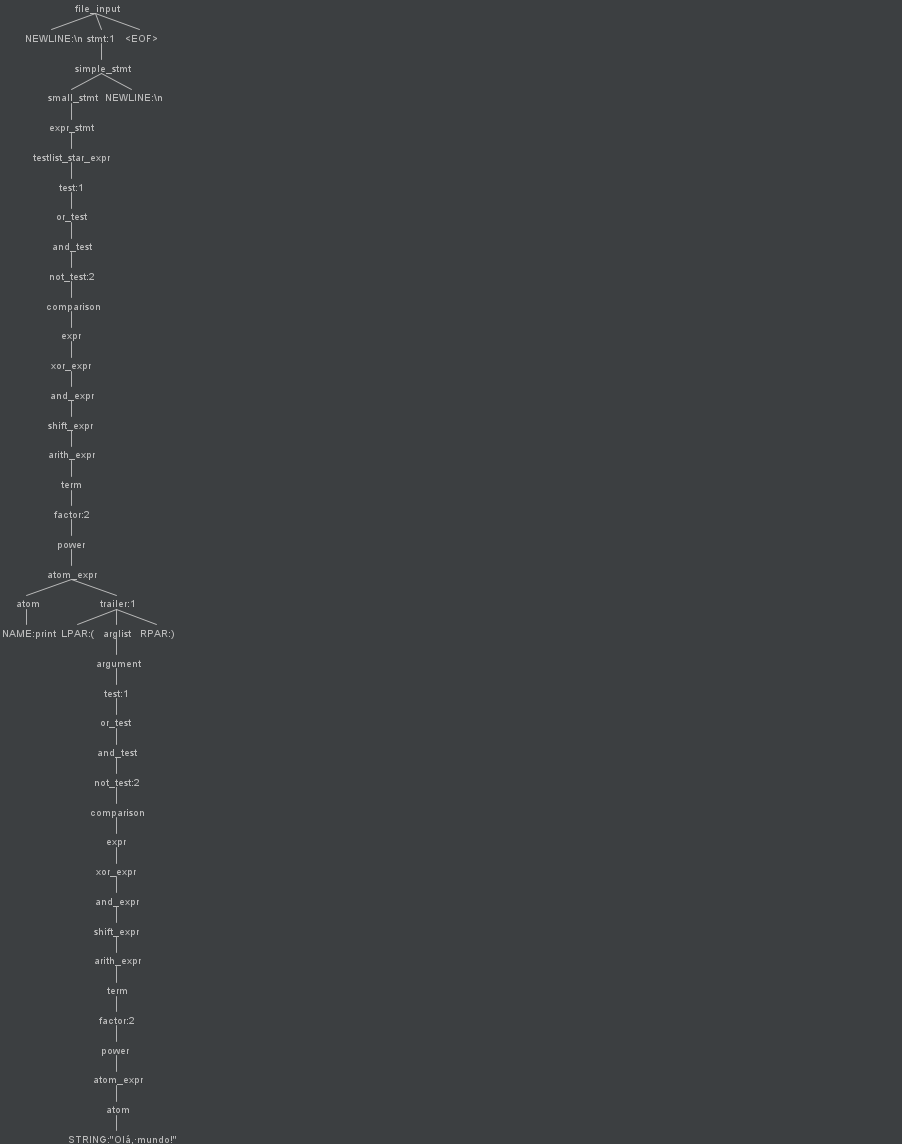
**Entradas aceitas:**

Entrada 1:

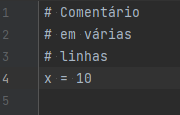
Comentário de uma linha



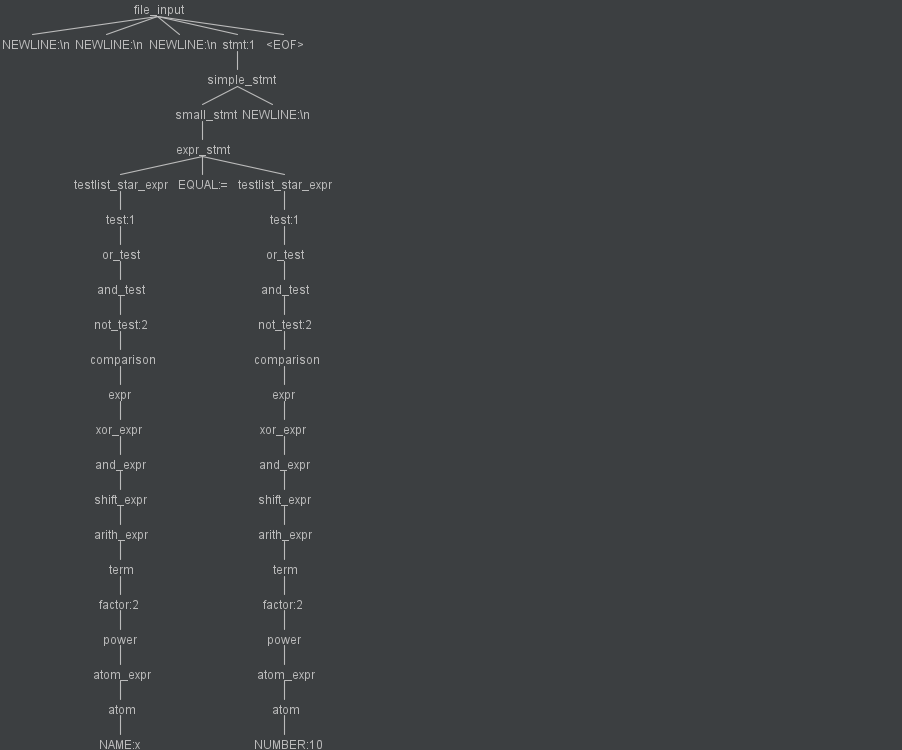
Nesse caso, a linha começando com # é considerada um comentário de uma linha e é ignorada pelo analisador léxico.



Entrada 2: Comentário em várias linhas



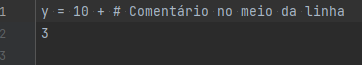
Neste exemplo, temos um comentário em várias linhas que começa com # em cada linha. Todas essas linhas são consideradas comentários e serão ignoradas durante a análise léxica.



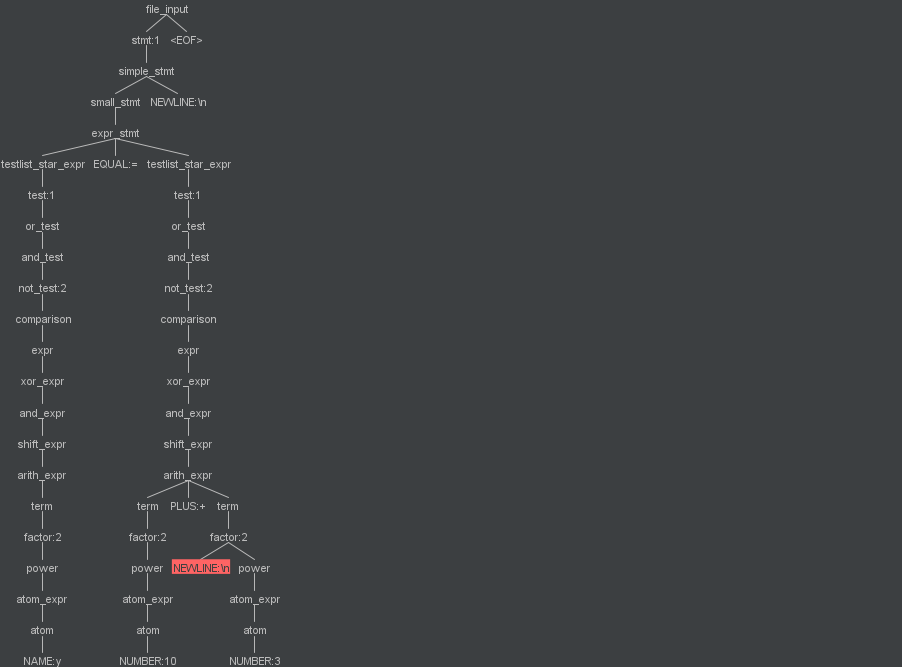
\*Em ambos os casos, os comentários são reconhecidos pela regra COMMENT e não são tratados como tokens no processo de análise léxica.

**Entradas rejeitadas:**

1. Comentário no meio da linha



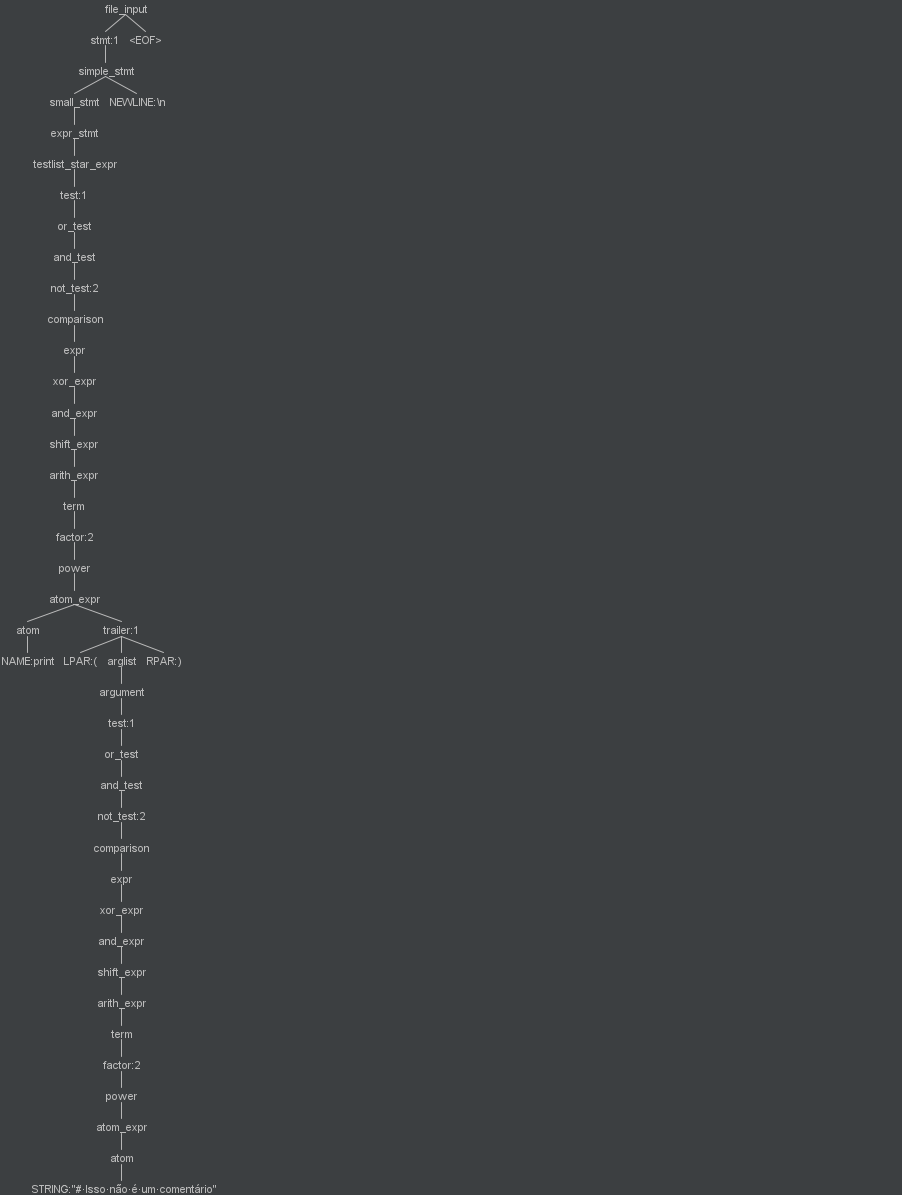
Nesse exemplo, o comentário foi colocado no meio de uma expressão matemática. A regra COMMENT não reconhecerá isso como um comentário válido, e o analisador léxico encontrará um token inválido, já que a quebra de linha após o + não é tratada adequadamente na regra de comentário.



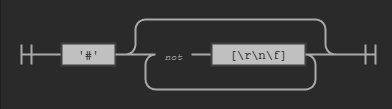
1. Comentário no meio de string



Neste exemplo, o texto entre aspas é uma string e não um comentário. A regra COMMENT não se aplicaria a isso, e o analisador léxico consideraria o conteúdo da string como parte do código em vez de ignorá-lo como um comentário.



**Diagrama de Sintaxe Equivalente**

****