

- Símbolos úteis (relevantes para as fases posteriores da compilação): são tokens, ou símbolos terminais:
  - Palavras-chave;
  - Operadores;
  - Símbolos de pontuação;
  - Identificadores;
  - Literais.
- Símbolos inúteis (relevantes apenas para melhorar a legibilidade do exto, sem significado para as fases posteriores da compilação):
  - Espaços em branco;
  - Tabulações;
  - Mudanças de linha;
  - Comentários.

# Token ::= ...

# Separator ::= ...

- O analisador léxico pode reconhecer e classificar todos os tokens de uma única vez, ou então reconhecer e classificar um por um, sob as ordens do analisador sintático;
- O analisador léxico analisa caracter por caracter do programa-fonte;
- Símbolos inúteis são reconhecidos e descartados;
- Com base no primeiro caracter de um símbolo útil, ele determina a regra que deve ser usada;
- Ele avança no programa-fonte conforme os caracteres estejam de acordo com a regra selecionada;
- Ele para quando encontrar um caracter que não faz parte da regra selecionada;
- Todos os caracteres analisados são classificados em TIPO + VALOR;
- TIPO é um código numérico que identifica o símbolo útil (é usado pelo analisador sintático);
- VALOR é uma cadeia de caracteres que identifica o símbolo reconhecido (é usado pelo analisador de contexto e pelo gerador de código).

**Formato Livre ou Fixo**

Numa linguagem de formato FIXO(menos comum), os separadores só podem ocorrer em posições pré-determinadas do programa-fonte.

Numa linguagem de formato LIVRE (mais usual), o analisador léxico deve reconhecer:

**Separator\* (Token Separator)\*,**

**OU**

**(Separator\* Token)\* Separator\***

**que são a mesma coisa**

ou seja:

- O uso de separadores é livre e serve para aumentar a legibilidade do programa-fonte;
- Uma seqüência de símbolos úteis, separados por zero ou mais símbolos inúteis;
- Símbolos inúteis também podem aparecer no início e/ou no final do programa;
- Eventualmente, um separador precisa ser usado para separar um símbolo útil de outro.