# CONTEXT-FREE GRAMMARS

→ Chomsky (Liguist) described four classes of generative devices or grammars that defines four classes of languages.

→ Two of these grammars classes

Context-free ReGular

→ These grammars turned out to be useful for describing the Syntax of Programming Language.

- Regular Grammar — Describes → Tokens } of P.L.
- Context-Free Grammar — Describe → Syntax } of P.L.

Basically

**Context-Free Grammar** is a Language Generator which describes Syntax of Natural Language.

# Backus-Naur Form

In 1960 John Backus and Peter Naur introduced formal notation method for describing Syntax of programming language which is known As Backus-Naur Form, or Simply BNF

* BNF was basically designed for ALGOL 60

BNF and context-free Grammars was nearly identical.

BNF is a metalanguage for programming languages.

Metalanguage is a language that is used to describe another language.

# BNF: Symbols to describe a Syntax

::= means 'is defined as'

< > means 'can be described as'

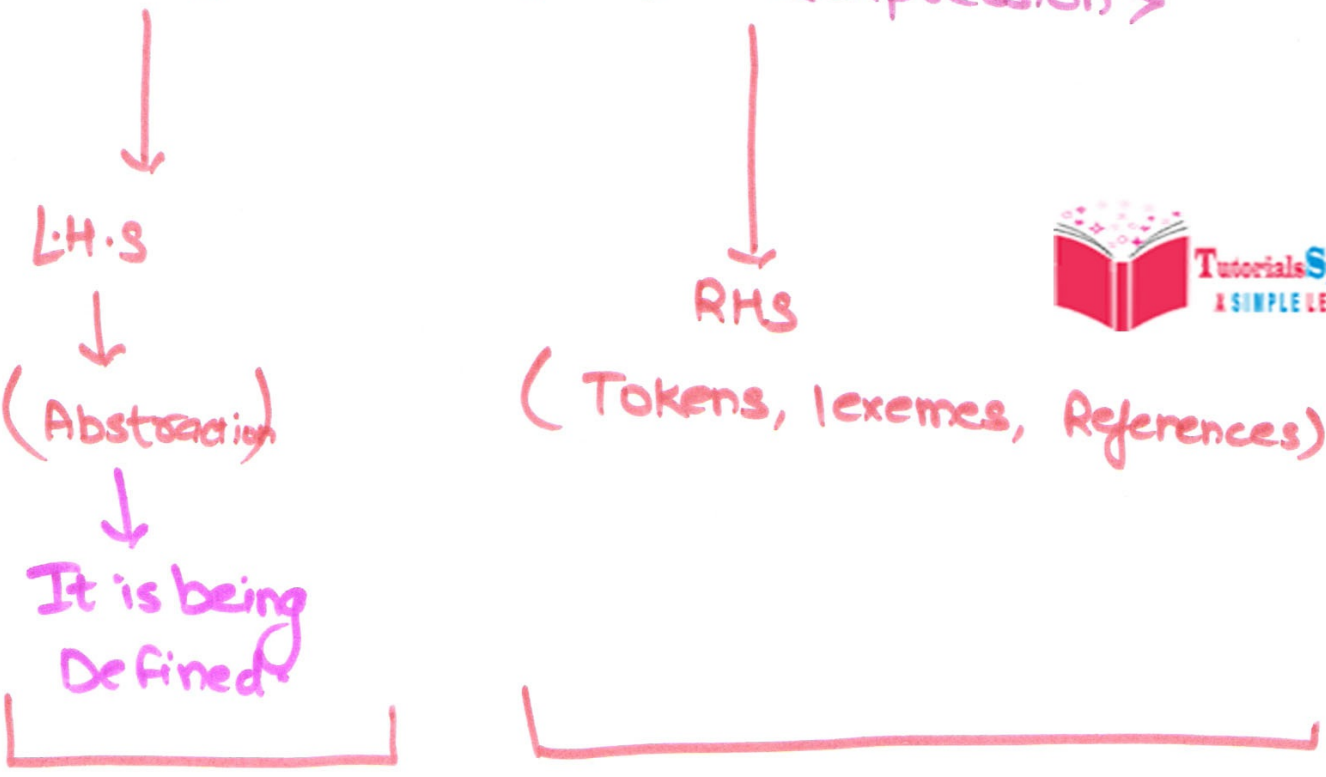
| means 'Or'



BNF uses "abstractions" for Syntatic Structures.

For Eg:- Java assignment Statement

<assign> ⟶ <var> = <expression>



**Terminals:** Symbols that can appear in the Output of a language because of its rules, but cannot be changed by rules themselves.

**Non-Terminals:** Syntatic entities that defines a part of the grammar.

Non-Terminals can be replaced and they are string, Composition of Terminals and Non-Terminals

**Eg. of Terminals**

+ = { } * 1

if while Do , int

double float A , B



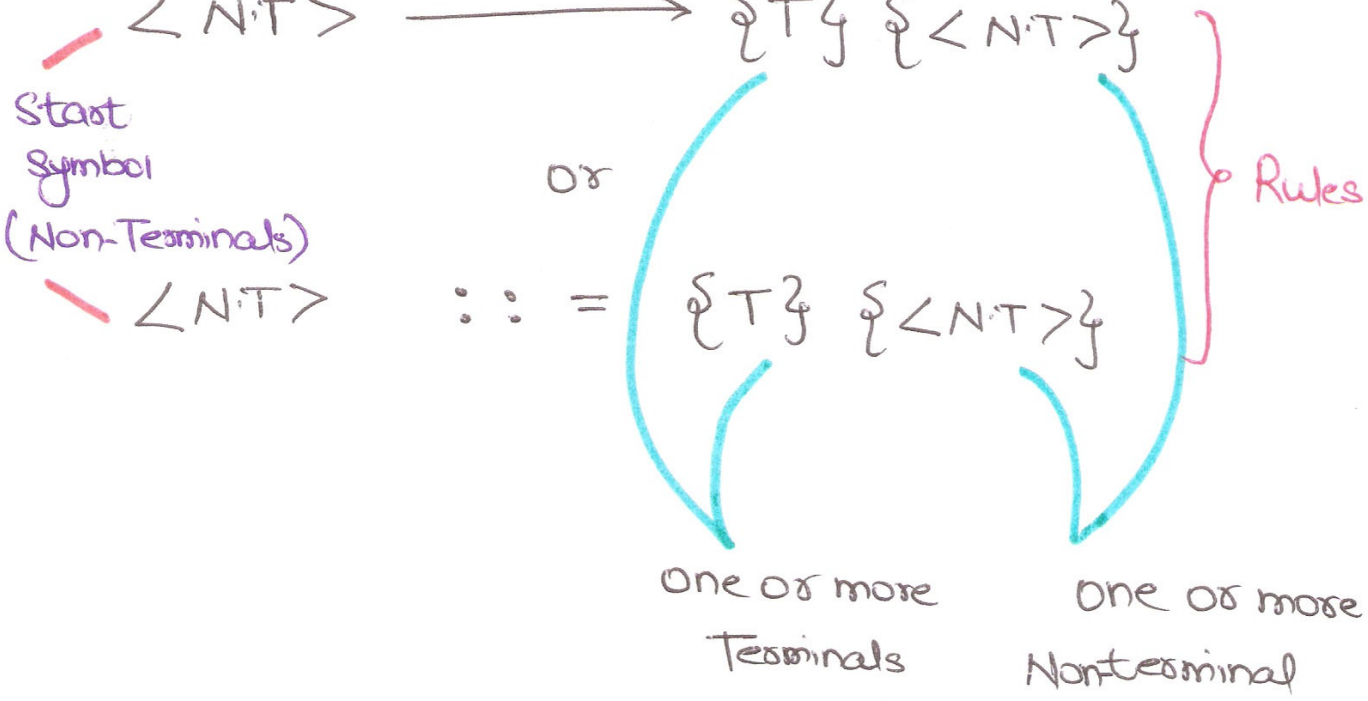
**Eg. of Non-Terminals:**

Sentences Expressions List of Numbers.

Words Statements

terms Programs

# How To write Rules in BNF




Example : if Loop

Structure ①

```
if ( condition )
{
   body
}
```

## BNF of Structure ①

<if_loop> ::= <if_part>

<if_part> ::= if (<condition>) {<body>}

or

<if_loop> ::= if (<condition>) {<body>}



## Structure ②

```
if (condition)
{ body
}
else
{
  body
}
```



## BNF of Structure ②

① <if_loop> ::= <if_Part> <else_Part>

<if_Part> ::= if (<condition>) {<body>}

<else_Part> ::= else {<body>}

or

② <if_loop> ::= if (<condition>) {<body>} else {<body>}

Example : Recursion in BNF