

Syntax.in

```
math_operator = "+" | "-" | "*" | "/" | "%" | "**"
relational_operator = "==" | "!=" | "<" | "<=" | ">" | ">="
boolean_operator = "and" | "or" | "not"
type = int | str | char | double | bool
list = "deflist" identifier "[" ":" type ";"
variable = "defvar" identifier ":" type {"," identifier ":" type} ";"
number = int | double
mathematical_expression = number {math_operator number}
relational_operand = identifier | int | double | mathematical_expression
relational_expression = relational_operand relational_operator
relational_operand
expression = (mathematical_expression|relational_expression) {boolean_operator
expression}
condition = expression relation expression

assignment = identifier "=" expression ";"
input_output_statement = "input" "(" (identifier|type) {"," (identifier|type) }
)" ";" | "print" "(" (identifier|type) {"," (identifier|type) } ")" ";"
simple_statement = assignment | input_output_statement
compound_statement = simple_statement {";" compound_statement}
statement = compound_statement | if_statement | loop_statement
if_statement = "if" condition "{" statement "}" ["else" "{" statement "}"]
loop_statement = "loop" expression "{" {statement} "}"
```