BeastLang v-0.1 Documentation

BeastLang Syntax Specification (v0.1 - Alpha)

☐ **In v-0.1 Alpha consider to write your code in the codes variable on line 115 with a sample code prebuilt in it.**

1. Make Statement

Purpose:

Used to declare variables (similar to let or const in JavaScript).

Syntax:

make <identifier> = <value>

Examples:

make username = "admin"

make age = 16

make isAdmin = true

Grammar:

make_stmt ::= "make" identifier "=" value “

identifier ::= [a-zA-Z_][a-zA-Z0-9_]*

value ::= string | number | boolean

string ::= '"' .*? '"'

number ::= [0-9]+

boolean ::= "true" | "false"

Limitations (Current Version):

Only string and numbers types are supported.

No support for complex types like arrays, objects, or functions yet.

No type inference or reassignment (make is immutable for now).

2.Print Statement

Purpose:

Used to print values to the console (for debugging or output).

Syntax:

Print <expression>

Examples:

print "Hello-Beast!"

print username

print age + 5

Grammar:

print_stmt ::= "print" "(" expression ")" ";"

expression ::= identifier | value | arithmetic

3. Arithmetic Operators

Purpose:

Supports basic arithmetic between two numeric variables or constants.

Supported Operators:

+, -, *, /

Syntax:

make result = <number | identifier> <operator> <number | identifier>

Examples:

make a = 1

make b = 20

make sum = a + b

print(sum)

Grammar:

arithmetic_expr ::= operand operator operand

operand ::= number | identifier

operator ::= "+" | "-" | "*" | "/"

Limitations:

Only two operands allowed in a single arithmetic expression

Valid: a + b

Invalid: a + b + c

Both operands must be declared variables or numeric literals.

No operator precedence; expressions are evaluated left to right.

No support for nested expressions or brackets.

Division by zero not handled yet (will crash the compiler).

Known Issues / To Be Implemented:

Type checking and error handling are still under development.

Scoping and block structures (if, while, function) are not yet implemented.

No support for comments or multiline strings.

The transpiler currently does not optimize or minify output.