Interpreter Report

Unary Operators

- “-”
    - Meaning: Returns the arithmetic negation of its operand.
    - Accepted operand types: float only.
    - Operand evaluation order: Evaluate right-hand side first.
    - Implicit type conversion: None. Non-float operands raise runtime errors.

- “!”
    - Meaning: Returns the logical negation of the operand’s truthiness.
    - Accepted operand types: Any type.
    - Operand evaluation order: Evaluate right-hand side first.
    - Implicit type conversion: Converts operand to boolean according to truthiness rules.

Binary Operators

- "+"
    - Meaning: Adds two numbers or concatenates two strings.
    - Accepted operand types: (float, float) or (string, string).
    - Operand evaluation order: Left then right.
    - Implicit type conversion: None. Mixed types raise runtime error.
- "-"
    - Meaning: Arithmetic subtraction.
    - Accepted operand types: (float, float).
    - Operand evaluation order: Left then right.
    - Implicit type conversion: None.

- "*"
    - Meaning: Arithmetic multiplication.
    - Accepted operand types: (float, float).
    - Operand evaluation order: Left then right.
    - Implicit type conversion: None.

- "/"
    - Meaning: Arithmetic division.
    - Accepted operand types: (float, float).

    - Operand evaluation order: Left then right.
    - Implicit type conversion: None. Division by zero raises runtime error.

- "=="
    - Meaning: Checks equality of two operands.
    - Accepted operand types: Any two operands.
    - Operand evaluation order: Left then right.
    - Implicit type conversion: Python-style equality (e.g., 0 == False is true).

- "!="
    - Meaning: Checks inequality of two operands.
    - Accepted operand types: Any two operands.
    - Operand evaluation order: Left then right.
    - Implicit type conversion: Same as "==".

- "<", "<=", ">", ">="
    - Meaning: Numerical comparison.
    - Accepted operand types: (float, float).
    - Operand evaluation order: Left then right.
    - Implicit type conversion: None. Non-float operands raise runtime error.

Truthiness / Falsiness

- Falsy values:
    - null ("None")
    - false
    - 0
    - "" (empty string)
    - [] (empty list)
    - () (empty tuple)

- Truthy values:
    - Everything else

Runtime Errors

The following runtime errors can occur:

- Unary Errors

    - Using - on a non-number
    - Using an unsupported unary operator
- Binary Errors
    - Using a non-+ operator on strings
    - Division by zero (/ with right operand = 0)
    - Using an unsupported operator on two floats
    - Using any operator (except == or !=) between mismatched operand types

Example Expressions

- Good Expression
    - ‘ (5 - 2) * 3 + 1 ’ evaluates to ‘10.0’
    - ‘ “hello” + “world” ’ evaluates to ‘helloworld’
- Bad Expression
    - ‘ “hello” + 5 ’ raises a runtime error unsupported binary operation between str and float types
    - ‘ 10 / 0 ’ raises a runtime error for dividing by zero