Language syntax and features – Yahav Yehoshua Bariah - 326295417

1. Type -

INTEGER: Whole numbers (e.g., -3, 0, 42)

BOOLEAN: True and False

2. Operations -

a. Arithmetic Operations (for INTEGERs)

- **Addition (+)**: Adds two integers together.
    - **Syntax**: a + b
    - **Example**: 5 + 3 results in 8.
- **Subtraction (-)**: Subtracts one integer from another.
    - **Syntax**: a - b
    - **Example**: 10 - 4 results in 6.
- **Multiplication (*)**: Multiplies two integers.
    - **Syntax**: a * b
    - **Example**: 7 * 3 results in 21.
- **Division (/) (integer division)**: Divides one integer by another and returns the quotient – (can't divide by zero).
    - **Syntax**: a / b
    - **Example**: 9 / 2 results in 4 (if using integer division).
- **Modulo (%)**: Returns the remainder of the division of one integer by another - (can't make modulo by zero).
    - **Syntax**: a % b
    - **Example**: 10 % 3 results in 1.

b. Boolean Operations

- **AND (&&)**: Returns true if both operands are true.
    - **Syntax**: a && b
    - **Example**: true && false results in false.
- **OR (||)**: Returns true if at least one of the operands is true.
    - **Syntax**: a || b
    - **Example**: false || true results in true.

- **NOT (!)**: Reverses the boolean value of the operand.
    - **Syntax**: !a
    - **Example**: !true results in false.

c. Comparison Operations

- **Equal to (==)**: Checks if two values are equal.
    - **Syntax**: a == b
    - **Example**: 5 == 5 results in true.
- **Not equal to (!=)**: Checks if two values are not equal.
    - **Syntax**: a != b
    - **Example**: 7 != 5 results in true.
- **Greater than (>)**: Checks if the left value is greater than the right value.
    - **Syntax**: a > b
    - **Example**: 8 > 3 results in true.
- **Less than (<)**: Checks if the left value is less than the right value.
    - **Syntax**: a < b
    - **Example**: 4 < 9 results in true.
- **Greater than or equal to (>=)**: Checks if the left value is greater than or equal to the right value.
    - **Syntax**: a >= b
    - **Example**: 6 >= 6 results in true.
- **Less than or equal to (<=)**: Checks if the left value is less than or equal to the right value.
    - **Syntax**: a <= b
    - **Example**: 2 <= 5 results in true

3. Functions -

- **Named function definitions:** Defines a function with a specific name that can be called multiple times.
    - **Syntax**: Defun 'name'(parameter*) -> body content
    - **Example**: Defun fun(n) -> (n==0) || n*fun(n-1) - (example for recursive function)
- **Anonymous functions (lambda expressions):** Creates a function without a name, typically used for short, single-use functions that are needed only in one place or context.
    - **Syntax**: Lambd parameter*.(body content)(value*)
    - **Example**: Lambd x.(Lambd y. (x + y)(2))(3) – results in 5.

4. Comment -

- Used to add brief explanations or notes for a single line of code.
    - **Syntax**: '#' body content
    - **Example**: # Hello world