# Syntax Design for JAM:

* Token separators: space,\t,\n,\r\n
* Comments will be \*\*
* Multiline comments: \*- code -\*
* Keywords: fn-function declaration,if,else,var,return,import,loop(while),forloop(for)
* Identifiers: An identifier begins with a letter (A–Z or a–z) or an underscore (\_), followed by any combination of letters, digits (0–9), or underscores.
* Literals: A sequence of one or more digits.
* Float: One or more digits, a decimal point, then one or more digits. (Optionally, you could add exponent notation later.)
* String: A sequence of characters enclosed in double quotes. Supports escape sequences
* Operators:+,-,\*,/,%,=, ==, !=, <, <=, >, >=,&&, ||, !
* Return type of function : **Arrow:** ->
* Delimeters:(),{},[],”,”,:,;,.
* Datatype:Int,Float,Bool,String,Void,
* Arrays:var arr:[Int]=[1,2,3,4],var p:(Int,Int)=(10,20);
* String array: var user: { String: String } = { "name": "Alice", "role": "admin" };
* Structure:struct p{ name:String,age:Int}

// Function to calculate factorial

fn factorial(n: Int) -> Int {

if n <= 1 {

return 1;

} else {

return n \* factorial(n - 1);

}

}

var result:Int = factorial(5);