專門用來解析和執行基於 S-expression 的語言。

### 程式結構

* 語法（Syntax）:
  + 定義了 S-expression 的基本結構，包括原子（ATOM）、列表以及引用（QUOTE）。
  + 原子可為符號（SYMBOL）、整數（INT）、浮點數（FLOAT）、字串（STRING）、NIL、T（true），或空列表（LEFT-PAREN RIGHT-PAREN）。
* 樹狀結構:
  + 使用樹狀結構來表達計算表達式，例如 (+ (- 5 6) 3 2) 表示為一棵樹，根據 LISP 語言的點對（dotted pair）表示法。
  + 這樹狀結構用於存儲並計算 S-expression。
* 指令集:
  + 定義了多個函數和指令，如 cons、car、cdr、list、define、加法（+）、乘法（\*）、等於（=）等，支持基本的數據操作和流程控制。
  + 指令如 if、cond、lambda 提供條件判斷和函數定義的能力。

### 功能說明

* 函數定義:
  + PutbackToken: 將 token 放回輸入流。
  + GetNextToken: 獲取下一個 token。
  + IsAtom: 判斷 token 是否為原子。
  + Addwait 和 NewwaitRP: 處理括號與列表結構的生成。
  + Type 和 TypeNlasttype: 類型判斷與錯誤處理。
  + Preorder: 前序遍歷輸出樹。
  + Quoter: 處理引用，不計算引用以下的節點。
* 操作數據結構:
  + NewTnode: 新增樹的節點。
  + CopyTree 和 CleanTree: 複製與清除樹。
  + Iseqv, IsatomT, Islist2, 等: 檢查節點的類型或值。
* 流程控制:
  + if, cond, lambda: 進行條件判斷和函數定義。
  + Clean\_environment: 清除環境中的變數定義。
  + ProceduretoIns 和 InstructionDefinition: 處理和定義指令。
* 錯誤處理與測試:
  + Argumentnum: 檢查函數參數的數量是否正確。
  + CheckifinDeList\_S 和 CheckifisFunctioninDeList: 檢查變量或函數是否已定義。