DSVision DSL Manual

Quick guide for the home page PDF

Why this DSL

- Action words match operations (insert/delete/search)
- Demo friendly: one DSL block = multiple Python lines + animations
- Extensible: add keywords in lexer/parser/interpreter once
- LLM friendly: natural language can be translated (see LLM Guide)

5-second starter

```
Sequential demo {
  init [1, 2, 3] capacity 5
  insert 10 at 1
  delete at 0
  search 3
}
```

Result: capacity slots visible; insert/delete/search animate in order

Structures & operations

- Sequential: init/batch_init, insert [at], delete at, search, capacity
- Linked: init, insert_head/tail, insert at, delete at, search
- Stack: init (capacity), push/pop/peek; auto expand animation when ful
- BST/AVL: insert/delete/search, traverse inorder|preorder|postorder|lev
- Huffman: build_text | build_numbers, show_codes, encode/decode

Good vs avoid

- ok BST myTree { insert 50; insert 30; search 30 }
- bad make a tree with some numbers (missing type/data)
- ok insert 10 at 2
- bad insert 10 at (missing index)

Errors are caught early: parser reports line/column for typos or bad arg

Working with LLM mode

- Skip syntax: ask LLM to draft DSL, then review before execute
- Precise animations: use DSL to lock order (stack expand, AVL rotatio
- Entry points: top nav buttons for DSL Manual / LLM Guide

Cheatline: Lexer -> Parser -> Interpreter -> OperationStep -> Animatio