Accessible Spreadsheet Component - One Page Summary

What it is

<y11n-spreadsheet> is a Lit 3.0 + TypeScript web component that provides an accessible spreadsheet-style grid in the browser. It implements WAI-ARIA grid semantics with keyboard-first interaction for data entry and calculation workflows.

What use cases it covers

Primary use case: embedding a spreadsheet-like UI in web apps where users need to navigate, edit, select ranges, and run formulas while preserving keyboard and screen-reader accessibility.

What it does

- Implements WAI-ARIA grid roles, roving tabindex, and aria-live announcements for active cell feedback.
- Supports Excel-like keyboard controls: arrows, Tab/Shift+Tab, Enter, Escape, Delete/Backspace, and Ctrl/Cmd shortcuts.
- Provides in-cell editing via Enter, double-click, or type-to-edit, with optional read-only mode.
- Evaluates formulas with cell refs, ranges, arithmetic, comparisons, string concatenation, and built-in functions.
- Allows custom formula extension through registerFunction(name, fn).
- Handles copy/cut/paste via TSV clipboard serialization with bounds-aware paste behavior.
- Uses virtual row rendering plus scroll spacers to keep large datasets responsive.

How it works

- Grid data is a sparse Map keyed by "row:col" (CellData stores rawValue, displayValue, and type).
- SelectionManager tracks anchor/head cells and normalized ranges for keyboard and pointer selection.
- FormulaEngine parses and evaluates formulas, then recalculates formula cells after edits or data changes.
- ClipboardManager converts selected ranges to TSV and parses pasted TSV into bounded cell updates.
- The component dispatches cell-change, selection-change, and data-change events for integration hooks.

How to run/use (minimal)

1. Install dependencies: npm install
2. Run the local demo/dev app: npm run dev
3. Register the component in your app (repo pattern): import './src/index.ts'
4. Place the element in markup: <y11n-spreadsheet rows="50" cols="26"></y11n-spreadsheet>
5. Provide data through setData(...) or the data property; listen for cell-change/selection-change/data-change events.
6. Required Node.js version: Not found in repo.
7. Published package install and production import instructions: Not found in repo.