TEXT Block Examples

Example 1 : Basic TEXT Blocks

This is a simple prose paragraph. The TEXT directive allows you to write explanatory text that will be typeset as regular paragraphs in the output document.

You can have multiple TEXT blocks in sequence. Each TEXT block becomes a separate paragraph with proper spacing.

Example 2 : Smart Quotes

The TEXT directive automatically converts straight quotes to "smart quotes" for proper typography. This includes both "double quotes" and 'single quotes'.

For example: "The quick brown fox" becomes properly formatted with opening and closing quotes.

Example 3 : Inline Mathematical Notation

You can include inline mathematical expressions in TEXT blocks. For example, the function f(n) = n^{}2 computes the square of n.

More complex expressions work too: The formula for the sum of squares is sum($i = 1$ to n) of i^{}2 = n(n+1)(2n+1)/6.

Example 4 : Mixed Content

Z notation provides powerful tools for formal specification. The $\forall$ quantifier allows us to express universal properties.

$\forall x : \mathbb{N} \bullet x \geq 0$
The above predicate states that all natural numbers are non-negative, which is true by definition of N.

Example 5 : Explaining Z Notation

When defining a function like $square : N \longrightarrow$ N, we're declaring a total function from natural numbers to natural numbers.

$$square : \mathbb{N} \longrightarrow \mathbb{N}$$
$$\forall n : \mathbb{N} \bullet square(n) = n * n$$

The axdef block gives the signature and the constraint that defines the function's behavior.

Example 6 : Long Explanations

The TEXT directive is particularly useful for providing detailed explanations of formal specifications. It allows you to write at length about the design decisions, invariants, and assumptions that underlie your formal models.

By interspersing TEXT blocks with formal notation, you create documents that are both mathematically precise and human-readable. This is essential for effective formal methods practice.

Example 7 : Technical Writing

In practice, specifications should include both formal definitions and prose explanations. The formal notation provides precision and enables automated checking with tools like fuzz. The prose provides context and motivation that help readers understand the purpose and design of the specification.