



Think before you code

Quit

# Topics

- document structure
  - data abstraction
    - optimal typography
      - potential complications

# Document structure

- provide non traditional sectioning
- use name spaced cross references
  - tables of contents must be able to adapt themselves<sup>▣</sup>
  - anything can and will be reused<sup>▣</sup>
  - many things can become a marginal, foot- or endnote<sup>▣</sup>
  - anything goes into headers, footers and statusinfo<sup>▣</sup>
  - there are more uses than we can think of

# Data abstraction

- think of future demands and tools
- there is a difference between local and global data
  - never code something more than once<sup>▣</sup>
  - force consistency by using abbreviations and alike
  - avoid hard coded names in changing documents<sup>▣</sup>

# Optimal typography

- it's in the small points, never compromise on forehand<sup>▣</sup>
- tables can be typeset differently, flows can come back<sup>▣</sup>
  - math, physical units and chemicals are not the same
  - languages have different demands and conventions
    - we seldom know in advance what it will look<sup>▣</sup>

# Potential complications

- lack of structure and consistency
- input, output and glyph encodings, forms and searching
  - use of periods, capitals, quotes and spaces