APL STYLES: ANTI-PATTERNS AND PATTERNS

Escaping the Beginner’s Plateau

Major Takeaway

Good APL code exposes the entire picture at a glance in a way that facilitates rapid manipulation, careful analysis, and minimal mental overhead. Not the big picture, not the abstracted details, but the entire picture in all its detail.

Experienced programmers, but novice APLers, will need to consciously counteract mainstream programming tendencies to “get” APL and to gain serious benefits from its use.

Success with Iverson style array programming is about a human experience more than a semantic robustness.

8 PATTERNS
VS. ANTI-PATTERNS

8 Patterns/Anti-Patterns

Iverson Style APL	Traditional Best Practices
Brevity	Verbosity
Macro	Micro
Transparency	Abstraction
Idioms	Libraries
Data	Control-Flow
Structure	Names
Implicit	Explicit
Syntax	Semantics

Thank you.

The virtues of APL that strike the programmer most sharply are its **terseness** — complicated acts can be described briefly, its **flexibility** — there are a large number of ways to state even moderately complicated tasks (the language provides choices that match divergent views of algorithm construction), and its **composability** — there is the possibility to construct sentences — one-liners as they are commonly called — that approach in the flow of phrase organization, sequencing and imbedding, the **artistic possibilities** achievable in **natural language prose.**

-- *Alan Perlis on APL One-liners and Lyrical Programming*

The human experience matters!