SUI MOVE

Security

Deterministic

Object-Centric

Resource Oriented

Syntax Feature	Example Syntax	Security Benefit
Only Unsigned Integers	let x: u64 = 10;	Prevents negative numbers and underflow; all overflows abort, no silent wrapping.
Strings as Vectors of Bytes	struct String { bytes: vector<u8> }	No hidden encoding bugs; no null-terminated strings; safe, explicit manipulation.
Vectors as Only Collection	let v: vector<u64> = vector[1, 2, 3];	Predictable memory layout; safe, bounds-checked access; no complex or unsafe data structures.
Explicit Type Annotations	let a: u8 = 1; let b: u64 = a as u64;	No silent type confusion or truncation; safer arithmetic and conversions.
No Nulls, Use Option Types	let x: Option<u64> = std::option::none();	No null pointer bugs; must explicitly handle absence of value, reducing runtime errors.
No Implicit Asset Creation	public fun mint(): Coin { ... }	All asset creation/destruction is explicit and auditable; prevents accidental or malicious mint/burn.
No Reentrancy by Design	(Static function calls, no dynamic dispatch)	Prevents reentrancy attacks by making all calls static and predictable.