Why Declare Variables?

An identifier like *Foo* in a TLA$^+$ formula can represent one of four kinds of entities: a variable, a constant (a parameter whose value is fixed throughout any behavior), a user-defined symbol, or a bound symbol introduced by a construct such as a quantifier. (Some user-defined symbols like *Nat* are defined in extended modules.) We indicate which kind *Foo* is by requiring it to be declared in a VARIABLE or CONSTANT statement if it is a variable or constant. (A user-defined symbol is effectively declared by its definition, and a bound symbol is declared by the construct that introduces it.)

In principle, we could eliminate declarations by using different colors to distinguish between variables, constants, and other symbols. However, the widespread use of black and white printing makes this impractical. We could also use disjoint sets of identifiers for these three kinds of symbols—for example, letting the identifier's initial letter determine which kind it is. However, experience with programming languages has shown that to be a bad idea. Hence, we use declarations.

CLOSE