Haskell uses types and classes.

A *type* is a collection of related values. For example, Bool is a type which can hold False and True. The correct notation for this is $v :: T$ to mean that v is a value in the type T. Every expression must have a type, which is calculated at runtime by *type inference.* The type of a function is determined by its inputs and outputs. For example, we would say
$(+) :: Numa => a- > a- > a$
because plus takes two Num inputs a and outputs a num a as well.