```
ASTNode  =  Stmt | Module | Expr

Mod      =  Module(ASTNode* body)

Stmt     =  FunctionDef(Identifier name, CompoundSmt body)

            | ClassDef(Indentifier name, Bases bases, ASTNode* body)

            | Return(Expr value)

            | Assign(Expr* targets, Expr value)

            | AugAssign(Expr target, Expr value)

            | For(CompoundStmt body)

            | While(Expr test, CompoundStmt body)

            | If(Expr test, CompoundStmt body, Stmt orelse)

            | Bases(Identifier* data)

            | Try(CompoundStmt body, Except except)

            | Except(CompoundStmt body)

            | Raise()

            | Import(Identifier* names)

            | Exec()

            | VariableDecl(Expr* right)

Expr     =  BinOp(Expr left, Expr right)

            | UnaryOp(Expr operand)

            | Comparison(Expr left, Expr* comparators)

            | Call(String func, String obj)
```