

Go 'syntax' Paketi Özeti

Fonksiyon / Tip	Açıklama	Örnek Kod	Notlar
ParseFile	Bir Go dosyasının/parçasını parse edip bir AST döndürür.	<code>*syntax.FileSyntax ParseFile("example.go", io.Reader veya dosya adı)</code>	Kaynak kodun bir Reader veya dosya adı ile açılması gerekir.
File	Bir .go dosyasının AST'ini temsil eder (paket adı, tüm importlar, fonksiyonlar/decl'ler).	<code>File</code>	AST'nin üst düzeyi.
Node	Tüm AST düğümlerinin ortak arayüzü.	Kullanıcılar genelde Walk ile görür.	Tür iddiaları ile (x.(*syntax.Node)).
Decl	Bildirim (declaration) türü: FuncDecl, ImportDecl, VarDecl, TypeDecl.	<code>for decl, _ := range f.DeclList { ... }</code>	Kaynak dosyadaki global bildirimler.
FuncDecl	Bir fonksiyon bildirimini temsil eder.	<code>if fn, ok := d.(*syntax.FuncDecl); ok { ... }</code>	Fonksiyonun adı, parametreleri, gövdesi.
ImportDecl	Bir import bildirimini temsil eder.	<code>imp := decl.(*syntax.ImportDecl)</code> <code>fmt.Println(imp.Path.Value)</code>	Import edilen paketlerin yolunu verir.
Expr	İfade düğümlerinin ana türü. Alt tipler: CallExpr, BasicLit, BinaryExpr, BoolLit, CharLit, ComplexLit, Ellipsis, FloatLit, IntegerLit, InterfaceType, MapType, PointerType, RangeExpr, SelectorExpr, SliceExpr, StructType, StructLit, UnaryExpr, ValueSpec, XBinaryExpr, XCallExpr, XCompositeLit, XIdent, XIndexExpr, XInterfaceType, XMapType, XPointerType, XRangeExpr, XSelectorExpr, XSliceExpr, XStructType, XStructLit, XUnaryExpr, XValueSpec.	<code>CallExpr</code> (fonksiyon çağrısı), <code>BasicLit</code> (sabitler).	İfadeleri çözümlemek için kullanılır.
CallExpr	Fonksiyon çağrıları temsil eder.	<code>fmt.Println("Çağrı:", call.Fun)</code>	<code>fmt.Println("...")</code> gibi ifadeler.
BasicLit	Sayılar, stringler gibi temel literal değerler.	<code>fmt.Println(123)</code>	Kod içindeki sabit değerler.
Stmt	Tüm ifade cümleleri (statement) için temel tür. Alt tipler: AssignStmt, BranchStmt, CaseClause, ForStmt, ForRangeStmt, IfStmt, ReturnStmt, SendStmt, SwitchClause, TypeSwitchClause, ValueSpec.	<code>AssignStmt</code> , <code>ForStmt</code> , <code>ReturnStmt</code> .	Kod akışını temsil eder.
Walk	AST üzerinde DFS dolaşım yapar.	<code>syntax.Walk(file, func(n syntax.Node) bool { ... })</code>	Tüm AST düğümlerini ziyaret etmek için kullanılır.
ParserMode	Parse seçenekleri: CheckBranches, AllowBareParses, AllowTypeParams, AllowGoChanges.	<code>syntax.ParseFile("example.go", nil, syntax.NewParser(syntax.ParserModeCheckBranches))</code>	Parse kuralları.