▼ *Package*	
• **package** *PackageName* **;** *ImportList S*	
▼ *PackageName*	
• *Name*	
• *Name*.*PackageName*	
▼ *ImportList*	
• *null*	
• **import** *PackageName* **;** *ImportList*	
▼ *S*	**statement**
• *null*	empty statement
• *E*	
• *E* **;** *S*	semi-separated exprs
▼ *E*	**expression**
• *E op E*	infix
• *E* **,** *E*	comma
• *U* **:=** *E*	update
• *A*	apply
• *D*	declare
• *I*	index
• *P*	prefix
• *R*	
• *X*	xml
▼ *op*	**infix operator**
• **+**	
• **-**	
• *	
• **	
• **mod**	
• **div**	
• **++**	**concatenation**
▼ *U*	**LHS of assignment**
• *Id*	
• *A*	
▼ *A*	**application**
• *E*.*Id*	postfix apply
• *Id* **(** *E* **)**	prefix apply
• *P* **(** *E* **)**	general prefix apply
▼ *D*	**definition**
• **var** *Id* **=** *E*	mutable
• **val** *Id* **=** *E*	immutable (was const)
• **function** *A* **=>** *S* **endfunction**	
▼ *I*	**index**
• *E* **[** *E* **]**	
• *E* **[** *E* **..** *E* **]**	
▼ *P*	**prefix expression**

  - *Literal*
  - *Id*
  - **(** *S* **)**
  - **{** *S* **}** list construction
  - *If*
  - *Switch*
  - *For*
- *Literal*
  - *String* e.g. "sna"
  - *Integer* e.g. -42
  - *Atom* e.g. `fu`
  - *Characters* e.g. 'bar'
- *If*
  - **if** *E ThenDo S CascadeList OptElse* **endif**
  - **unless** *E ThenDo S Cascade OptElse* **endunless**
- *ThenDo*
  - **do**
  - **then**
- *CascadeList*
  - *null*
  - **elseif** *E ThenDo S CascadeList*
  - **elseunless** *E ThenDo S CascadeList*
- *OptElse*
  - *null*
  - **else** *S*
- *Switch*
  - **switch** *E CaseList OptElse* **endswitch**
- *CaseList*
  - *null*
  - **case** *E ThenDo S CaseList*
- *For*
  - **for** *ForList ThenDo S OptFinally* **endfor**
- *ForList*
  - *null*
  - *BindingList*
  - *ConditionList*
  - *BindingList* **;** *ConditionList*
- *BindingList*
  - *Binding*
  - *Binding* **;** *BindingList*
- *Binding* **generator**
  - *Id* **times** *E*
  - *Id* **in** *E*
  - *Id* **&** *Id* **in** *G* *a design error!*
  - *Id* **from** *E*

• *Id* **from** *E* **to** *E*	
• *Id* **from** *E* **by** *E* **to** *E*	
▼ *ConditionList*	
• *Condition OptionalBreak*	
• *Condition OptionalBreak* **;** *ConditionList*	
▼ *Condition*	**condition**
• **while** *E*	
• **until** *E*	
▼ *OptionalBreak*	
• *null*	
• **break** *E*	
▼ *OptFinally*	
• *null*	
• **finally** *S*	
▼ *R*	**relational chain**
• *E relop E*	
• *E relop R*	
▼ *relop*	**relational operator**
• **=**	structural equality
• **==**	identity
• **<**	numerical less than
• **>**	numerical greater than
• **<=**	numerical less than or equal
• **>=**	numerical greater than or equal
▼ *X*	**XML element**
• **<** *Name AttributeList* **>** *E* **</** *Name* **>**	
• **<** *Name AttributeList* **/>**	
▼ *AttributeList*	
• *null*	
• *Name* = *P AttributeList*	