

<i>function</i>	\Rightarrow	$\begin{aligned} & \backslash\mathbf{begin} \left\{ \mathbf{eqcode} \right\} \left\{ \mathbf{id} \right\} \\ & \quad \left\{ [id [, id]^*] \right\} \\ & \quad \left\{ [ext_type [, ext_type]^*] \right\} \left\{ ext_type \right\} \\ & \quad instr_list \\ & \quad \backslash\mathbf{end} \left\{ \mathbf{eqcode} \right\} \end{aligned}$
<i>indexes</i>	\Rightarrow	$([upper] [lower] \mid lower\ upper)$
<i>idx</i>	\Rightarrow	$\mathbf{id} \ indexes$
<i>numx</i>	\Rightarrow	$(\mathbf{num} \mid divide) \ indexes$
<i>idx_numx</i>	\Rightarrow	$(idx \mid numx)$
<i>upper</i>	\Rightarrow	$\begin{aligned} & \wedge \left\{ ([\ expr] \mid expr) \right\} \\ & \mid \\ & \wedge (\mathbf{id} \mid \mathbf{num}) \end{aligned}$
<i>linear</i>	\Rightarrow	$\begin{aligned} & \mathbf{id} [(+ \mid -) \mathbf{num}] \\ & \mid \\ & \mathbf{num} \end{aligned}$
<i>lower</i>	\Rightarrow	$\begin{aligned} & - \left\{ expr [, expr]^* \right\} \\ & \mid \\ & - (\mathbf{id} \mid \mathbf{num}) \end{aligned}$
<i>type</i>	\Rightarrow	$\backslash\mathbf{type} \left\{ (\mathbf{Z} \mid \mathbf{R} \mid \mathbf{N} \mid \mathbf{B}) \right\}$
<i>ext_type</i>	\Rightarrow	$\begin{aligned} & type [\wedge (\left\{ sexpr \right\} \mid \mathbf{num} \mid \mathbf{id}) \\ & \quad [- (\left\{ sexpr [, sexpr]^* \right\})] \mid \mathbf{id} \mid \mathbf{num}) \end{aligned}$
<i>instr_list</i>	\Rightarrow	$[instr \ \backslash\mathbf{lend}]^*$
<i>instr</i>	\Rightarrow	$\begin{aligned} & assign \\ & \mid \\ & declare \\ & \mid \\ & with_loop \\ & \mid \\ & comment \\ & \mid \\ & if_cond \\ & \mid \\ & return \end{aligned}$
<i>if_cond</i>	\Rightarrow	$\begin{aligned} & \backslash\mathbf{qif} \left\{ cond_block \right\} \\ & \quad instr_list \\ & \quad [\backslash\mathbf{qelseif} \left\{ cond_block \right\} \\ & \quad \quad instr_list]^* [\backslash\mathbf{qelse} \\ & \quad \quad instr_list]^* \backslash\mathbf{qendif} \end{aligned}$
<i>cond_block</i>	\Rightarrow	$\begin{aligned} & expr [comp \ expr]^+ \\ & \quad [set_op \ expr [comp \ expr]^+]^* \end{aligned}$
<i>assign</i>	\Rightarrow	$idx [generator] \ \backslash\mathbf{gets} \ expr$
<i>declare</i>	\Rightarrow	$idx \ \backslash\mathbf{in} \ ext_type$

<i>boolop</i>	\Rightarrow	$\backslash\text{land}$ $\backslash\text{lor}$ $\backslash\text{oplus}$
<i>binop</i>	\Rightarrow	$+$ $-$ $\backslash\text{cdot}$ $\backslash\text{ll}$ $\backslash\text{gg}$ $\backslash\text{mod}$
<i>divide</i>	\Rightarrow	$(\backslash\text{frac} \mid \backslash\text{dfrac}) \{ \text{expr} \} \{ \text{expr} \}$
<i>function_call</i>	\Rightarrow	$\backslash\text{call} \{ \text{id} \} \{ [\text{expr} [, \text{expr}]^*] \}$
<i>sexpr</i>	\Rightarrow	$(\backslash\text{not} \mid -) \text{sexpr_op} [(\text{binop} \mid \text{boolop}) \text{sexpr_op}]^*$ (expr) $\{ \text{expr} \}$
<i>sexpr_op</i>	\Rightarrow	$(\text{idx_numx} \mid \text{function_call} \mid \text{vector} \mid \text{matrix})$
<i>filter</i>	\Rightarrow	$\backslash\text{filter} \{ \text{id} \wedge \{ [\text{id}] \} \}$ $[, \text{id} \wedge \{ [\text{id}] \}]^*$ $\text{generator} \}$
<i>genarray</i>	\Rightarrow	$\backslash\text{genar} \backslash\text{limits} \wedge \{ \text{expr} \} (\text{expr})$
<i>vector</i>	\Rightarrow	$\backslash\text{begin} \{ \text{tvector} \}$ $[\text{expr} \backslash\text{lend}]^+$ $\backslash\text{end} \{ \text{tvector} \}$
<i>matrix</i>	\Rightarrow	$\backslash\text{begin} \{ \text{tmatrix} \} \{ \text{id} \}$ $[\text{expr} [\& \text{expr}]^* \backslash\text{lend}]^+$ $\backslash\text{end} \{ \text{tmatrix} \}$
<i>expr</i>	\Rightarrow	$(\text{sexpr} \mid \text{filter} \mid \text{genarray}) \text{indexes}$
<i>with_loop</i>	\Rightarrow	$\text{idx} \mid \text{generator} \backslash\text{gets} (\text{expr} \mid \text{with_loop_cases})$
<i>with_loop_cases</i>	\Rightarrow	$\backslash\text{begin} \{ \text{cases} \}$ $[\text{expr} \& \text{generator}]^+$ $[\text{expr} \& \backslash\text{otherwise}]^+$ $\backslash\text{end} \{ \text{cases} \}$

<i>return</i>	\Rightarrow	<code>\return { <i>expr</i> }</code>
<i>generator</i>	\Rightarrow	<code>\forall id [, id]*</code>
		<code>id [, id]* : cond_block</code>
<i>comp</i>	\Rightarrow	<code><</code>
		<code>></code>
		<code>\leq</code>
		<code>\geq</code>
		<code>[\not] =</code>
<i>set_op</i>	\Rightarrow	<code>(\cup \cap)</code>