function	$\Rightarrow$	**\begin** **{** **eqcode** **}** **{** ***id*** **}**
		{ *[id [, id]*]* **}**
		{ *[ext_type [, ext_type]*]* **}** **{** *ext_type* **}**
		instr_list
		\end **{** **eqcode** **}**
indexes	$\Rightarrow$	*([upper] [lower] | lower upper)*
idx	$\Rightarrow$	***id*** *indexes*
numx	$\Rightarrow$	*(* ***num*** *| divide) indexes*
idx_numx	$\Rightarrow$	*(idx | numx)*
upper	$\Rightarrow$	^ **{** *([(linear | expr)]	expr)* **}**
	|	^ *(* ***id*** *|* ***num*** *)*
linear	$\Rightarrow$	*(\iter [(+ | -)* ***num*** *] |* ***num*** *)*
lower	$\Rightarrow$	_ **{** *expr [, expr]** **}**
	|	_ *(* **id** *|* **num** *)*
type	$\Rightarrow$	**\type** **{** *(* **Z** *|* **R** *|* **N** *|* **B** *)* **}**
ext_type	$\Rightarrow$	*type [^ (* **{** *sexpr* **}** *|* ***num*** *|* ***id*** *)*
		[_ (**{** *sexpr [, sexpr]** **}** *]] |* ***id*** *|* ***num*** *)*
instr_list	$\Rightarrow$	*[instr* **\lend** *]**
instr	$\Rightarrow$	*assign*
	|	*declare*
	|	*index_loop*
	|	*comment*
	|	*if_cond*
	|	*return*
if_cond	$\Rightarrow$	**\qif** **{** *cond_block* **}**
		instr_list
		[**\qelseif** **{** *cond_block* **}**
		instr_list]* [**\qelse**
		instr_list]* **\qendif**
cond_block	$\Rightarrow$	*expr [comp expr]+*
		[set_op expr [comp expr]+]*
assign	$\Rightarrow$	*idx [generator]* **\gets** *expr*
declare	$\Rightarrow$	*idx* **\in** *ext_type*

$$boolop \Rightarrow \textbf{\textbackslash land}$$
$$| \quad \textbf{\textbackslash lor}$$
$$| \quad \textbf{\textbackslash oplus}$$

$$binop \Rightarrow \textbf{+}$$
$$| \quad \textbf{--}$$
$$| \quad \textbf{\textbackslash cdot}$$
$$| \quad \textbf{\textbackslash ll}$$
$$| \quad \textbf{\textbackslash gg}$$
$$| \quad \textbf{\textbackslash mod}$$

$divide \Rightarrow$ (**\frac** | **\dfrac**) { *expr* } { *expr* }

$function\_call \Rightarrow$ **\call** { ***id*** } { *[expr [* **,** *expr]** * *]* }

$sexpr \Rightarrow$ (**\lnot** | **--**) *sexpr_op [(binop | boolop) sexpr_op]** *
$| \quad$ (*expr*)
$| \quad$ { *expr* }

$sexpr\_op \Rightarrow$ (***idx_numx*** | *function_call* | *matrix*)

$filter \Rightarrow$ **\filter** { ***id*** ^ { [***id***] }
[**,** ***id*** ^ { [***id***] } *]** *
| *generator* }

$genarray \Rightarrow$ **\genar** **\limits** ^ { *expr* } (*expr*)

$matrix \Rightarrow$ **\begin** { **tmatrix** }
[expr [**&** *expr]** * **\lend** *]+*
\end { **tmatrix** }

$expr \Rightarrow$ (*sexpr* | *filter* | *genarray*) *indexes*

$index\_loop \Rightarrow$ *idx* **|** *generator* **\gets** (*expr* | *index_loop_cases*)

$index\_loop\_cases \Rightarrow$ **\begin** { **cases** }
[expr **&** *generator]+*
[expr **&** **\otherwise** *]+*
\end { **cases** }

$$
\begin{array}{lcl}
\textit{return} & \Rightarrow & \textbf{\textbackslash return} \quad \{ \quad \textit{expr} \quad \} \\
\textit{generator} & \Rightarrow & \textbf{\textbackslash forall} \quad \textbf{\textit{id}} \quad \textit{[} \quad \textbf{,} \quad \textbf{\textit{id}} \quad \textit{]}^{*} \\
& | & \textbf{\textit{id}} \quad \textit{[} \quad \textbf{,} \quad \textbf{\textit{id}} \quad \textit{]}^{*} \quad \textbf{:} \quad \textit{cond\_block} \\
\textit{comp} & \Rightarrow & \textbf{<} \\
& | & \textbf{>} \\
& | & \textbf{\textbackslash leq} \\
& | & \textbf{\textbackslash geq} \\
& | & \textit{[} \, \textbf{\textbackslash not} \quad \textit{]} \, \textbf{=} \\
\textit{set\_op} & \Rightarrow & \textit{(} \quad \textbf{\textbackslash cup} \quad | \quad \textbf{\textbackslash cap} \quad \textit{)}
\end{array}
$$