

- **newline:**
Newlines are combined, so a single newline is the same as multiple.
- **module-name:**
Modules are flags given to the interpreter/compiler, to let it know you want to be using certain rules, functions, or features. Available modules: `FORS` (randomness), `FRACTIO` (fractions), `MAGNVM` (large integers), `SCRIPTA` (file I/O: `LEGE`, `SCRIBE`, `ADIVNGE`), `SVBNVLLA` (negative literals), `RETE` (networking: `PETE`, `PETITVR`, `AVSCVLTA`).
- **id:**
Variable. Can only consist of lowercase characters and underscores, but not the letters `j`, `u`, or `w`.
- **builtin:**
Builtin functions are uppercase latin words.
- **string:**
Any text encased in `"` or `'` characters. Single-quoted strings are always literal. Strings support 1-based indexing (`string[II]`) and inclusive slicing (`string[II VSQVE IIII]`), returning single-character strings and substrings respectively.
- **interpolated-string:**
A double-quoted string containing `{expression}` segments. Each expression is evaluated and coerced to a string. Use `{` and `}` for literal braces.
- **numeral:**
Roman numerals consisting of the uppercase characters `I`, `V`, `X`, `L`, `C`, `D`, and `M`. Can also include underscore if the module `MAGNVM`.
- **bool:**
`VERITAS` or `FALSITAS`.
- **binop:**
Binary operators: `+`, `-`, `*`, `/`, `RELIQVVM` (modulo), `EST` (equality), `DISPAR` (not-equal), `MINVS` (`<`), `PLVS` (`>`), `HAVD_PLVS` (`≤`), `HAVD_MINVS` (`≥`), `ET` (and), `AVT` (or), `&` (string concatenation), `@` (array concatenation).
- **unop:**
Unary operators: `-` (negation), `NON` (boolean not).

Top-level		
<i>program</i>	→	<i>optional-newline module-calls statements</i>
<i>module-calls</i>	→	<i>module-call newline module-calls</i>
<i>module-calls</i>	→	
<i>module-call</i>	→	CUM module-name
<i>statements</i>	→	<i>statement newline statements</i>
<i>statements</i>	→	
<i>optional-newline</i>	→	newline
<i>optional-newline</i>	→	
Statements		
<i>statement</i>	→	<i>expression</i>
<i>statement</i>	→	DESIGNA id VT <i>expression</i>
<i>statement</i>	→	DESIGNA id , ids VT <i>expression</i>
<i>statement</i>	→	id AVGE <i>expression</i>
<i>statement</i>	→	id MINVE <i>expression</i>
<i>statement</i>	→	DEFINI id (<i>optional-ids</i>) VT <i>scope</i>
<i>statement</i>	→	<i>if-statement</i>
<i>statement</i>	→	DVM <i>expression</i> FAC <i>scope</i>
<i>statement</i>	→	AETERNV FAC <i>scope</i>
<i>statement</i>	→	PER id IN <i>expression</i> FAC <i>scope</i>
<i>statement</i>	→	PER id, id-list IN <i>expression</i> FAC <i>scope</i>
<i>statement</i>	→	DONICVM id VT <i>expression</i> VSQVE <i>expression</i> FAC <i>scope</i>
<i>statement</i>	→	REDI (<i>optional-expressions</i>)
<i>statement</i>	→	ERVMPE
<i>statement</i>	→	CONTINVA
<i>statement</i>	→	<i>try-statement</i>
<i>try-statement</i>	→	TEMPTA <i>scope</i> CAPE id <i>scope</i>
<i>if-statement</i>	→	SI <i>expression</i> TVNC <i>scope</i>
<i>if-statement</i>	→	SI <i>expression</i> TVNC <i>scope</i> <i>optional-newline else-statement</i>
<i>else-statement</i>	→	ALIVD <i>scope</i>
<i>else-statement</i>	→	ALIVD <i>if-statement</i>
<i>scope</i>	→	<i>optional-newline { newline statements }</i>
Expressions		
<i>expression</i>	→	(<i>expression</i>)
<i>expression</i>	→	id
<i>expression</i>	→	builtin (<i>optional-expressions</i>)
<i>expression</i>	→	INVOC <i>expression</i> (<i>optional-expressions</i>)
<i>expression</i>	→	FVNCTIO (<i>optional-ids</i>) VT <i>scope</i>
<i>expression</i>	→	<i>literal</i>
<i>expression</i>	→	<i>expression</i> [<i>expression</i>]
<i>expression</i>	→	<i>expression</i> [<i>expression</i> VSQVE <i>expression</i>] (inclusive slice)
<i>expression</i>	→	<i>expression</i> binop <i>expression</i>
<i>expression</i>	→	unop <i>expression</i>
<i>literal</i>	→	string
<i>literal</i>	→	interpolated-string
<i>literal</i>	→	numeral
<i>literal</i>	→	bool
<i>literal</i>	→	[<i>optional-expressions</i>]
<i>literal</i>	→	[<i>expression</i> VSQVE <i>expression</i>] (inclusive on both ends)
<i>literal</i>	→	TABVLA { <i>optional-dict-items</i> }
<i>optional-dict-items</i>	→	<i>dict-items</i>
<i>optional-dict-items</i>	→	
<i>dict-items</i>	→	<i>expression</i> VT <i>expression</i> , <i>dict-items</i>
<i>dict-items</i>	→	<i>expression</i> VT <i>expression</i>
Lists		
<i>optional-ids</i>	→	<i>ids</i>
<i>optional-ids</i>	→	
<i>ids</i>	→	id, ids
<i>ids</i>	→	id
<i>optional-expressions</i>	→	<i>expressions</i> 2
<i>optional-expressions</i>	→	
<i>expressions</i>	→	<i>expression, expressions</i>
<i>expressions</i>	→	<i>expression</i>