

- **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).
- **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.
- **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` (`>`), `ET` (and), `AVT` (or), `&` (string concatenation).
- **unop:**  
Unary operators: `-` (negation), `NON` (boolean not).

<b>Top-level</b>		
<i>program</i>	→	<i>optional-newline module-calls statements</i>
<i>module-calls</i>	→	<i>module-call <b>newline</b> module-calls</i>
<i>module-calls</i>	→	
<i>module-call</i>	→	<b>CUM module-name</b>
<i>statements</i>	→	<i>statement <b>newline</b> statements</i>
<i>statements</i>	→	
<i>optional-newline</i>	→	<b>newline</b>
<i>optional-newline</i>	→	
<b>Statements</b>		
<i>statement</i>	→	<i>expression</i>
<i>statement</i>	→	<b>DESIGNA id</b> VT <i>expression</i>
<i>statement</i>	→	<b>DESIGNA id , ids</b> VT <i>expression</i>
<i>statement</i>	→	<b>id AVGE</b> <i>expression</i>
<i>statement</i>	→	<b>id MINVE</b> <i>expression</i>
<i>statement</i>	→	<b>DEFINI id</b> ( <i>optional-ids</i> ) VT <i>scope</i>
<i>statement</i>	→	<i>if-statement</i>
<i>statement</i>	→	<b>DVM</b> <i>expression</i> <b>FACE</b> <i>scope</i>
<i>statement</i>	→	<b>AETERNVM</b> <b>FACE</b> <i>scope</i>
<i>statement</i>	→	<b>PER id</b> <b>IN</b> <i>expression</i> <b>FACE</b> <i>scope</i>
<i>statement</i>	→	<b>DONICVM id</b> VT <i>expression</i> <b>VSQVE</b> <i>expression</i> <b>FACE</b> <i>scope</i>
<i>statement</i>	→	<b>REDIC</b> ( <i>optional-expressions</i> )
<i>statement</i>	→	<b>ERVMPE</b>
<i>statement</i>	→	<b>CONTINVA</b>
<i>if-statement</i>	→	<b>SI</b> <i>expression</i> <b>TVNC</b> <i>scope</i>
<i>if-statement</i>	→	<b>SI</b> <i>expression</i> <b>TVNC</b> <i>scope</i> <i>optional-newline else-statement</i>
<i>else-statement</i>	→	<b>ALVID</b> <i>scope</i>
<i>else-statement</i>	→	<b>ALVID</b> <i>if-statement</i>
<i>scope</i>	→	<i>optional-newline { <b>newline</b> statements }</i>
<b>Expressions</b>		
<i>expression</i>	→	( <i>expression</i> )
<i>expression</i>	→	<b>id</b>
<i>expression</i>	→	<b>builtin</b> ( <i>optional-expressions</i> )
<i>expression</i>	→	<b>INVoca</b> <i>expression</i> ( <i>optional-expressions</i> )
<i>expression</i>	→	<b>FVNCTIO</b> ( <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> <b>VSQVE</b> <i>expression</i> ] (inclusive slice)
<i>expression</i>	→	<i>expression</i> <b>binop</b> <i>expression</i>
<i>expression</i>	→	<b>unop</b> <i>expression</i>
<i>literal</i>	→	<b>string</b>
<i>literal</i>	→	<b>interpolated-string</b>
<i>literal</i>	→	<b>numeral</b>
<i>literal</i>	→	<b>bool</b>
<i>literal</i>	→	[ <i>optional-expressions</i> ]
<i>literal</i>	→	[ <i>expression</i> <b>VSQVE</b> <i>expression</i> ] (inclusive on both ends)
<i>literal</i>	→	<b>TABVLA</b> { <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>
<b>Lists</b>		
<i>optional-ids</i>	→	<i>ids</i>
<i>optional-ids</i>	→	
<i>ids</i>	→	<b>id</b> , <i>ids</i>
<i>ids</i>	→	<b>id</b>
<i>optional-expressions</i>	→	<i>expressions</i>
<i>optional-expressions</i>	→	
<i>expressions</i>	→	<i>expression</i> , <i>expressions</i>
<i>expressions</i>	→	<i>expression</i> 2