

Syntax Specification for Naturally

Kushagra Mehta

2013A7PS006H

Formatting and Characters

<indent> = '\n\t'

<space> = ' '

<newline> = '\n'

<specialchar> = '\'|~'|!'|@'|#'|\$'|%'|^'|&'|*'|(|')|_|'-'|+'|='|{|}|'|[|']|\"'|'|'|'|'|'|';|:|'|<|'|.|'|>|/|'|?'

<digit> = [0-9]

<alphabet> = [a-z A-Z]

<sign> = +|-|e

Comments

<comment> = '//'<string>

Data Types

<boolean> = 'True' | 'False'

<string> = <alphabet><string> | <specialchar><string> | <digit><string> |

<alphabet> | <digit> | <specialchar>

<integer> = <sign><digit><integer> | <digit>

<float> = <sign><integer>.<integer>

<list> = [] | [<value>,<listnext>] | [<value>]

<listnext> = <value><listnext> | e

<dictionary> = {} | {<value>:<value>} | {<value>:<value>,<dictnext>}

<dictnext> = <value>:<value>,<dictnext> | <value>:<value>

<value> = <string> | <integer> | <float> | <list> | <dictionary> | <boolean>
| <functionvalue>
<valuelist> = <value><space>,<space><valuelist> | <value>

Identifiers

<identifier> = [a-z A-Z][a-z A-Z 0-9]*
<identifierlist> = <identifier><space>,<space><identifierlist> | <identifier>

Variable Declarations

'Let' <identifier> 'be' <value> |
'Let' <identifier> 'be' <identifier> |
'Set' <identifier> 'to' <value> |
'Set' <identifier> 'to' <identifier> |
'Declare' <identifier> 'as' <value> |
'Declare' <identifier> 'as' <identifier> |
'Define' <identifier> 'as' <value> |
'Define' <identifier> 'as' <identifier> |

Functions

<functiondefinition> = <functiondeclaration><functionbody>
<functiondeclaration> = 'declare' <space> 'function' <space>
<identifier><indent> |
'declare' <space> 'function' <space> <identifier> <space> 'with' <space>
'arguments' <space> <identifierlist><indent>
<functionbody> = <program><returnstmt><functionbody> | e
<returnstmt> = 'return' <space> <identifierlist> <newline>
<functionvalue> = 'call' <space> <identifier> <space> 'with' <space>
<identifierlist> |
'value' <space> 'from' <space> <identifier> <space> 'with' <space>
<identifierlist> |

'value' <space> 'from' <space> <identifier> <space> 'with' <space>
<identifierlist>

Expressions and Operators

<booleanexpression> = (<booleanexpression>) | <expression>
<booleanoperator> <expression> | <boolean>
<expression> = (<expression>) | <expression> <operator> <expression> |
<value>
<operator> = '**' |
 '/' | '*' | '%' |
 '+' | '-' |
 '>=' | '>' | '<=' | '<' |
 '==' |
 '='
<booleanoperator> = '>=' | '>' | '<' | '<=' | '=='

I/O

<outputstmt> = 'print' <identifierlist> | 'print' <value>
<inputstmt> = 'input' <identifier> | 'read' <identifier>

Conditional

<conditional> = 'if' <space> <booleanexpression> <indent> <program>
 <elifblock>
 <elseblock>
<elifblock> = 'elif'
<space> <booleanexpression> <indent> <program> <elifblock> |
 'else' <space> 'if' <space>
<booleanexpression> <indent> <program> <elifblock> | e
<elseblock> = 'else' <indent> <program> | e

Loop

<loop> = 'while' <space> <booleancondition> <indent> <program>

Program Definition

<program> = <comment> <newline> <program> |
 <functiondefinition> <newline> <program> |
 <variabledeclaration> <newline> <program> |
 <expression> <newline> <program> |
 <outputstmt> <newline> <program> |
 <inputstmt> <newline> <program> |
 <conditional> <newline> <program> |
 <loop> <newline> <program> |
 <newline> <program> |
e