Cheatsheet: Pseudocode

Assignment

Variables	Identifier ← Exp
Constant	**constant** Identifier ← Exp

Comments

single line comment
Another comment

Input/Output

User Input	**USERINPUT**
Output	**OUTPUT** StringExp

Iteration (Condition-Controlled)

Repeat-Until	**REPEAT** #some code **UNTIL** BoolExp
While	**WHILE** BoolExp #some code **ENDWHILE**

Arithmetic

Real Numbers (float, double)	**+ - * /**
Integer Division	IntExp **DIV** IntExp
Integer Modulus	IntExp **MOD** IntExp

Relational Operators

Less than	Exp < Exp
Greater than	Exp > Exp
Equal to	Exp = Exp
Not equal to	Exp ≠ Exp
Less than or equal to	Exp ≤ Exp
Greater than or equal to	Exp ≥ Exp

Boolean Operations

AND	BoolExp **AND** Bool Exp
OR	BoolExp **OR** Bool Exp
NOT	**NOT** Bool Exp

Iteration (Count-Controlled)

For	**FOR** Identifier ← IntExp **TO** IntEx #some code **UNTIL** BoolExp

Key

Exp is an expression

Bold highlights key terms

Type Conversion

String to Integer	**STRING_TO_INT(**StringExp**)**
String to Real	**STRING_TO_REAL(**StringExp**)**
Integer to String	**INT_TO_STRING(**IntExp**)**
Real to String	**REAL_TO_STRING(**RealExp**)**

Selection

If	**IF** BoolExp **THEN**
	# statements here
	ENDIF

If-Else	**IF** BoolExp **THEN**
	# code to do if true
	ELSE
	# code to do else
	ENDIF

Else-If	**IF** BoolExp **THEN**
	# code to do if true
	ELSE IF BoolExp **THEN**
	# code if this true
	ELSE
	# code to do else
	ENDIF

Arrays

Assignment	Identifier ← **[**Exp, Exp, …, Exp**]**
Accessing an Element	Identifier**[**IntExp**]**
Updating an element	Identifier**[**IntExp**]** ← Exp
Array Length	**LEN**(Identifier)

Two-Dimensional Arrays

Accessing an Element	Identifier**[**IntExp**] [**IntExp**]**
Updating an element	Identifier**[**IntExp**] [**IntExp**]** ← Exp

Random Number Generation

Generate random number between two integers	**RANDOM_INT**(IntExp, IntExp)

String Handling

String Length	**LEN(**StringExp**)**
Position of a character	**POSITION(**StringExp,CharExp**)**
Concatenation	StringExp **+** StringExp

Substrings are created by the first parameter that gives the start position, the second parameter gives the final position and the third parameter is the string itself…

SUBSTRING(IntExp, IntExp, StringEXp)

Subroutines

Define a subroutine	**SUBROUTINE** Identifier**(**parameters**)** # some code to run **ENDSUBROUTINE**
Returning a value	**SUBROUTINE** Identifier**(**parameters**)** # some code to run **RETURN** Exp **ENDSUBROUTINE**
Calling a subroutine	Identifier**(**parameters**)**