

IB Pseudocode Syntax

Basic instructions

Name	Syntax	Description	Examples
Assign	<i>variable = value</i>	Assigns a value to the variable	a = 1
Input	input <i>variable</i>	Inputs the variable	input a
Output	output <i>variable/expression</i>	Outputs a value of the variable or expression	output a output "hello" output 2 + 2
Create	create <i>type variable</i>	Creates the variable with standard value of given type	create Boolean a create Number b create String c
Delete	delete <i>variable</i>	Deletes the variable	delete a

Conditions

Name	Syntax	Description	Examples
If	if <i>condition</i>	Indicates the start of a condition block and states the first condition	if a = 1
Else if	else if <i>condition</i>	States an additional condition	else if a = 2
Else	else	Indicates the start of the part of a condition block which will be executed if all conditions above are False	else
End if	end if	Indicates the end of a condition block	end if

Loops

Name	Syntax	Description	Examples
While loop	loop while <i>condition</i>	Executes a loop block while the condition is true	loop while a < 5
Until loop	loop until <i>condition</i>	Executes a loop block until the condition is true	loop until a = 5
For loop	loop <i>variable</i> from <i>start value</i> to <i>end value</i>	Executes a loop block for every value of the variable between start value and end value	loop a from 1 to 5
End loop	end loop	Indicates the end of a loop block	end loop

Operators

Name	Syntax	Description	Examples
Equal	<i>value1 = value2</i>	Checks is the first value equal to the second	a = 1
Not equal	<i>value1 != value2</i> <i>value1 <> value2</i>	Checks is the first value not equal to the second	a != 1 a <> 1
Greater	<i>value1 > value2</i>	Checks is the first value greater than the second	a > 1
Greater or equal	<i>value1 >= value2</i>	Checks is the first value greater or equal to the second	a >= 1
Less	<i>value1 < value2</i>	Checks is the first value less than the second	a > 1
Less or equal	<i>value1 <= value2</i>	Checks is the first value less or equal to the second	a >= 1
Not	NOT <i>value1</i>	Executes logical or bitwise NOT for the value	NOT a
And	<i>value1</i> AND <i>value2</i>	Executes logical or bitwise AND for the first and the second values	a AND 1
Or	<i>value1</i> OR <i>value2</i>	Executes logical or bitwise OR for the first and the second values	a OR 1
Xor	<i>value1</i> XOR <i>value2</i>	Executes bitwise XOR for the first and the second values	a XOR 1
Addition	<i>value1 + value2</i>	Adds the first and the second values	a + 1
Subtraction	<i>value1 - value2</i>	Subtracts the first and the second values	a - 1
Multiplication	<i>value1 * value2</i>	Multiplies the first and the second values	a * 1
Division	<i>value1 / value2</i>	Divides the first and the second values	a / 1
Modulo	<i>value1</i> mod <i>value2</i>	Gets modulo of the first and the second values	a mod 1
Integer division	<i>value1</i> div <i>value</i>	Gets integer part of the division of the first and the second values	a div 1

Functions

Name	Syntax	Description	Examples
Function	function <i>name</i> (<i>arg1</i> , ...)	Indicates the start of a function block with name and arguments	function f(a, b)
Return	return <i>variable/expression</i>	Returns value or expression from function	return a return "hello" return 2 + 2
End function	end function	Indicates the end of a function block	end function
Run function	<i>name</i> (<i>arg1</i> , ...)	Runs a function block with given name and arguments	f(1, 2)

Procedures

Name	Syntax	Description	Examples
Procedure	procedure <i>name</i> (<i>arg1</i> , ...)	Indicates the start of a procedure block with name and arguments	procedure p(a, b)
End procedure	end procedure	Indicates the end of a procedure block	end procedure
Run procedure	<i>name</i> (<i>arg1</i> , ...)	Runs a procedure block with given name and arguments	p(1, 2)

Basic data types

Name	Syntax	Description	Examples
Boolean	TRUE/FALSE create Boolean <i>variable</i>	Boolean type that can contain only TRUE or FALSE values	a = TRUE b = FALSE create Boolean c
Number	<i>number</i> create Number <i>variable</i>	Number type that can contain any number value	a = 0 b = 1 create Number c
String	" <i>string</i> " create String <i>variable</i>	String type that can contain any text	a = "hello" create String b

Arrays

Name	Syntax	Description	Examples
Create array	create Array <i>name</i>	Creates an empty array with given name	array a
Get item	<i>name</i> [<i>index</i>]	Returns an item with given index from the array	a[0]
Set item	<i>name</i> [<i>index</i>] = <i>value</i>	Assigns a value to given index from the array	a[0] = 1
Array size	<i>name</i> .size()	Return a size of the array	a.size()
Assign array	<i>name</i> = [<i>val1</i> , <i>val2</i> , ...]	Assigns an array with given values to a variable	a = [1, 2, 3]

Dictionaries

Name	Syntax	Description	Examples
Create dictionary	create Dictionary <i>name</i>	Creates an empty dictionary with given name	dictionary a
Get item	<i>name</i> [<i>key</i>]	Returns an item with given key from the dictionary	a["a"]
Set item	<i>name</i> [<i>key</i>] = <i>value</i>	Assigns a value to given key from a dictionary with given name	a["a"] = 1
Assign dictionary	<i>name</i> = { <i>key1</i> : <i>val1</i> , <i>key2</i> : <i>val2</i> , ...}	Assigns a dictionary with given values to a variable	a = {"a": 1, "b": 2}

Collections

Name	Syntax	Description	Examples
Create collection	create Collection <i>name</i>	Creates an empty collection with given name	collection a
Add item	<i>name</i> .addItem(<i>value</i>)	Adds a value to the end of the collection	a.addItem(1)
Get next	<i>name</i> .getNext()	Returns next value from the collection	a.getNext()
Reset next	<i>name</i> .resetNext()	Resets next element of the collection	a.resetNext()
Has next	<i>name</i> .hasNext()	Checks does the collection have next element	a.hasNext()
Is empty	<i>name</i> .isEmpty()	Check does the collection contains elements	a.isEmpty()
Collection size	<i>name</i> .size()	Return a size of the collection	a.size()

Stacks

Name	Syntax	Description	Examples
Create stack	create Stack <i>name</i>	Creates an empty stack with given name	stack a
Push	<i>name</i> .push(<i>value</i>)	Adds a value to the stack	a.push(1)
Pop	<i>name</i> .pop()	Gets a value from the stack	a.pop()
Is empty	<i>name</i> .isEmpty()	Check does the stack contains elements	a.isEmpty()
Stack size	<i>name</i> .size()	Return a size of the stack	a.size()

Queues

Name	Syntax	Description	Examples
Create queue	create Queue <i>name</i>	Creates an empty queue with given name	queue a
Enqueue	<i>name</i> .enqueue(<i>value</i>)	Adds a value to the queue	a.enqueue(1)
Dequeue	<i>name</i> .dequeue()	Gets a value from the queue	a.dequeue()
Is empty	<i>name</i> .isEmpty()	Check does the queue contains elements	a.isEmpty()
Queue size	<i>name</i> .size()	Return a size of the queue	a.size()