Jinx Core Library

# Using the Core Library

The Jinx core library contains a number of functions that are generally useful, such as outputting debug text and manipulating collections. To use the core library in a Jinx script, you must add the following text to the beginning of your script:

import core

-- other script code follows...

# Core Functions

## function write {params}

The write function sends text to the registered debug out function. It will attempt to convert any parameter type to a string, and will iterate over the values of a collection, sending each value to the debug out function in turn.

## function write line {params}

Operates similarly to the write function, but sends a newline character when finished.

## function {param} (get) size

Returns the count or size of param as an integer for supported types, otherwise returns null.

* Collection – Returns the number of elements.
* String – Returns the number of Unicode characters (not bytes).
* Buffer – Returns the number of bytes in the buffer.

## function {params} (is) empty

Returns true or false for supported types, otherwise returns null.

* Collection – Returns true if empty, false if not.
* String – Returns true if empty, false if not.
* Buffer – Returns true if empty, false if not.

## function {iterator param} (get) key

Returns collection element’s key for iterator param, and returns null for all other types or if element is not in collection.

## function {param} (get) value

Returns collection element’s value for iterator param, or the return value of a coroutine. Returns null for all other types or if element is not in collection.

## function (get) call stack

Returns collection of strings indicating the current call stack. The first entry is always “root”.

## function call {function param}

Invokes a function param, and returns the function’s return value.

## function call {function param} with {params}

Invokes a function param, forwarding all params to the function, and returns the function’s return value.

## function async call {function param}

Invokes an asynchronous function param, and returns a coroutine handle.

## function async call {function param} with {params}

Invokes an asynchronous function param, forwarding all params to the function, and returns a coroutine handle.

## function {coroutine param} (is) finished

Executes the coroutine param and returns true if finished.

## function all (of) {collection param} (are) finished

Executes all coroutines and returns true if all are finished.

## function any (of) {collection param} (is) finished

Executes all coroutines and returns true any is finished.

# Core Properties

## public readonly newline is "\n"

The public property newline is primarily intended for use as a parameter for the write or write line functions.