

## The Jack OS API

The Jack language comes with a collection of eight built-in classes that extend the language's capabilities. This standard library can be viewed as a basic operating system. This document gives the OS API.

### Math

A library of commonly used mathematical functions.

f c **multiply** returns the product of    and   . When a Jack compiler detects the multiplication operator '    ' in the program's code, it handles it by invoking this method. In other words, the Jack expressions    and Ma    return the same value.

f c **divide** returns the integer part of   . When a Jack compiler detects the division operator '    ' in the program's code, it handles it by invoking this method. In other words, the Jack expressions    and Ma d de    return the same value.

f c **min** returns the minimum of    and   .

f c **max** returns the maximum of    and   .

f c **sqr** returns the integer part of the square root of   .

### String

The String class represents character strings. In addition for constructing and disposing strings, the class features methods for getting and setting individual characters of the string, for erasing the string's last character, for appending a character to the string's end, and more typical string-oriented operations.

c c S **new** a Le constructs a new empty string with a maximum length of a Le and initial length of 0.

e d **dispose** disposes this string.

e d **length** returns the current length of this string.

e d c a **charAt** returns the character at the -th location of this string.

e d d **setCharAt** c a c sets the character at the -th location of this string to c.

e d S **appendChar** c a c appends c to this string's end and returns this string.

e d d **eraseLastChar** : erases the last character from this string.

e d **intValue** returns the integer value of this string, until a non-digit char is detected.

e d d **setInt** a sets this string to hold a representation of the given value.

f c c a **backSpace** returns the backspace character.

f c c a **doubleQuote** returns the double quote (    ) character.

f c c a **e L e** returns the newline character.

## Array

Represents an array. In the Jack language, arrays are instances of the `Array` class. Once declared, the array entries can be accessed using the usual syntax `a[i]`. Each array entry can hold a primitive data type or any object type. Different array entries can have different data types.

`f c        A a new        e` constructs a new array of the given size.

`e d        d dispose` disposes this array.

## Output

A library of functions for displaying text on the screen.

The Hack physical screen consists of 512 rows of 256 pixels each. The library uses a fixed font, in which each character is displayed within a frame which is 11 pixels high (including 1 pixel for inter-line spacing) and 8 pixels wide (including 2 pixels for inter-character spacing). The resulting grid accommodates 23 rows (indexed 0..22, top to bottom) of 64 characters each (indexed 0..63, left to right). The top left character position on the screen is indexed (0,0). A cursor, implemented as a small filled square, indicates where the next character will be displayed.

`f c        d moveCursor` moves the cursor to the `-th` column of the `-th` row, and erases the character displayed there.

`f c        d printChar c a c` displays the given character at the cursor location, and advances the cursor one column forward.

`f c        d printString S` displays the given string starting at the cursor location, and advances the cursor appropriately.

`f c        d printInt` displays the given integer starting at the cursor location, and advances the cursor appropriately.

## Keyboard

This class allows reading inputs from a standard keyboard.

`f c c a keyPressed` returns the character of the currently pressed key on the keyboard; if no key is currently pressed, returns 0. Recognizes all ASCII characters, as well as the following keys: `e e` (128=`S e e`), `bac ace` (129=`S bac ace`), `ef a` (130), `a` (131), `a` (132), `d a` (133), `e` (134), `e d` (135), `a e` (136), `a e d` (137), `e` (138), `de e e` (139), `ESC` (140), `F F` (141-152).

`f c c a readChar` waits until a key is pressed on the keyboard and released, then echoes the key to the screen and returns the character of the pressed key.

`f c S readLine S e a e` displays the message on the screen, reads from the keyboard the entered text until a `e e` character is detected, echoes the text to the screen, and returns its value. Also handles user backspaces.

`f c readInt S e a e` displays the message on the screen, reads from the keyboard the entered text until a `e e` character is detected, echoes the text to the screen, and returns its integer value (until the first non-digit character in the entered text is detected). Also handles user backspaces.

## Memory

This library provides two services: direct access to the computer's main memory (RAM), and allocation and recycling of memory blocks. The Hack RAM consists of 32,768 words, each holding a 16-bit binary number.

`f c peek add e` returns the RAM value at the given address.

`f c d poke add e a e` sets the RAM value at the given address to the given value.

`f c A a alloc e` finds an available RAM block of the given size and returns a reference to its base address.

`f c d deAlloc A a` de-allocates the given object (cast as an array) by making it available for future allocations.

## Sys

A library that supports various program execution services.

`f c d halt` halts the program execution.

`f c d error e C de` displays the given error code in the form "ERR e C de ", and halts the program's execution.

`f c d wait d a` waits approximately `d a` milliseconds and returns.