

About

The Quicksand Javascript Editor is an Integrated Development Environment (IDE) for programming small javascript games on a phone. Several games are included.

License: GPL

Github: github.com/quicksandjavascript

Hello World

At the top of the editor, create this code and click Run:

```
function helloworld(){  
  L.canvas(W/2).icon(127759,W/2,W/2);  
}
```

Run

Runs the game which contains the cursor.

Stop game with Run or Enter.

Save

Saves to a timestamped file.

The file contains your games AND the Editor.

Zoom/Fullscreen

Toggles font size.

Doubleclick toggles fullscreen.

Pretty Print

Displays a readonly version of code that can be saved as PDF.

Comment

Toggles comments for the current line.

Also puts the line in Paste.

Complete

Completes a word.

Looks for match in current game and then elsewhere.

Rename (find/replace)

To rename, select text, type text, click Paste several times. To skip, click Arrow.

Undo/Redo

Restore up to 50 changes.

Paste

Selects text and puts it in Paste. Once Paste is set, it pastes.

Backspace deletes selected text.

Enter moves the cursor to the end of selection and unselects.

Undo resets Paste.

Selection looks at the character to the right and then selects for:

```
newline: line  
characters: left,right  
spaces: right  
blank line: next blank line  
blocks: (){}[]""
```

Misc

Color-Picker and Icon-Picker put a code in Paste.

Edit-the-Editor enables changes to the editor source code, such as colors, new buttons, etc.

Swipeup

Insert capital letter, instead of using shift.

Errors

Runtime errors display an error message and automatically jump to the error in code.

Global Variables and Aliases

```
FPS=(frames per second)  
W=innerWidth;  
floor=Math.floor;
```

Overridden

Array sort: compares numbers instead of strings.

Variable tracking

```
L.debug(string,canvas)
```

Random number

```
number=L.rnd(max,min)
```

Random biased sets (reduces repetition)

```
set=L.RandomArray(quantity,range)  
set.next()
```

Sound

```
sound=L.Sound()  
sound.rocket()
```

Canvas

```
canvas=L.canvas(fontsize,iconstart)  
canvas.icon(emoji,x,y)  
canvas.white()
```

Color gradient

```
canvas.fillStyle=L.gradient(canvas,bottom,height,[[stop,color],...])
```

Shapes

```
canvas.fill(L.shape([[x,y,...],...],scale))
```

Pixel to Vector

```
canvas.fill(L.shape(L.vertices(pixels)))
```

Buttons

⬇ Save

▷ Run

📄 Pretty

📖 Help

◇ Misc

Ⓐ Zoom

← Left

→ Right

↶ Undo

↷ Redo

⌞ Shift

// Comment

⇒ Complete

📋 Paste

↵ Enter

⌫ Backspace