Debugging Node.js as you never seen before

Mary Marchini

@mmarkini

sthima

500 Internal Server Error

nginx/1.1.19

NETFLIX

Netflix Error

We were unable to process your request.

Please go to the Netflix home page by clicking the button below.

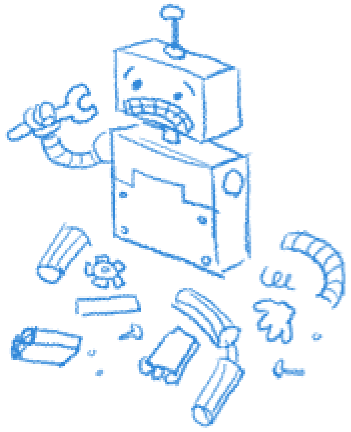
Netflix Home

Google

500. That's an error.

The server encountered an error and could not complete your request.

If the problem persists, please report your problem and mention this error message and the query that caused it. That's all we know.



Mary Marchini

Developers Team Leader @ **Sthima**

@mmarkini

http://mmarchini.me/

oss@mmarchini.me

https://github.com/mmarchini/

Tales of an Upgrade

Tales of an Upgrade

- Python v2.5
- Framework v2
- Cloud

Tales of an Upgrade

• Python v2.5	→	• Python v2.7
• Framework v2	→	• Framework v3
• Cloud	→	• Bare Metal

Tales of an Upgrade

• Python v2.5	→	• Python v2.7
• Framework v2	→	• Framework v3
• Cloud	→	• Bare Metal
		+ Features
		+ New Module

Tales of an Upgrade

Tales of an Upgrade

How to investigate bugs in production?

How to investigate bugs in production?

- Logging

How to investigate bugs in production?

- Logging
- Tracing

How to investigate bugs in production?

- Logging
- Tracing
- Core files

How to investigate bugs in production?

- Logging
- Tracing
- **Core files**

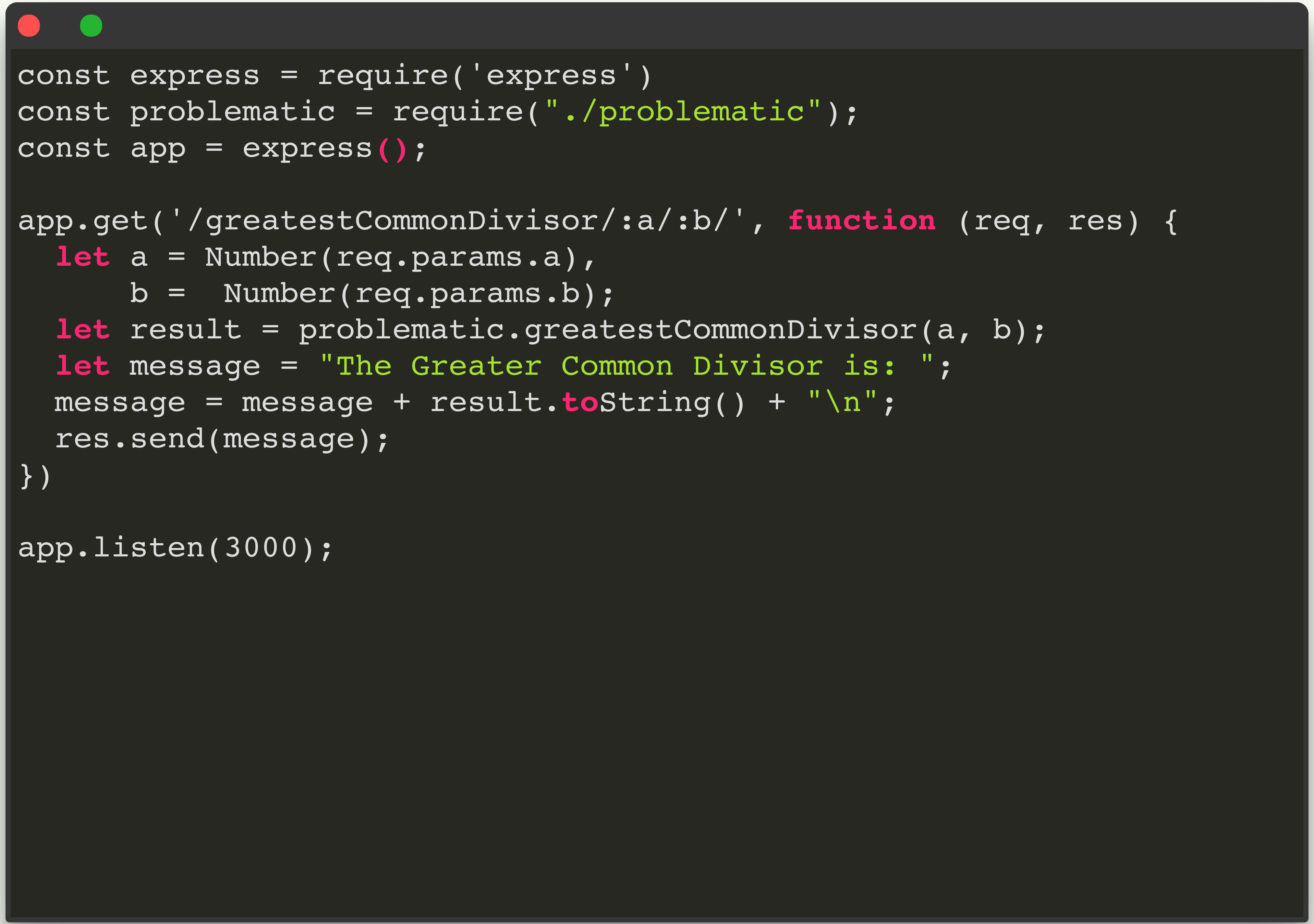
Core File Analysis

Core File Analysis

with lldb

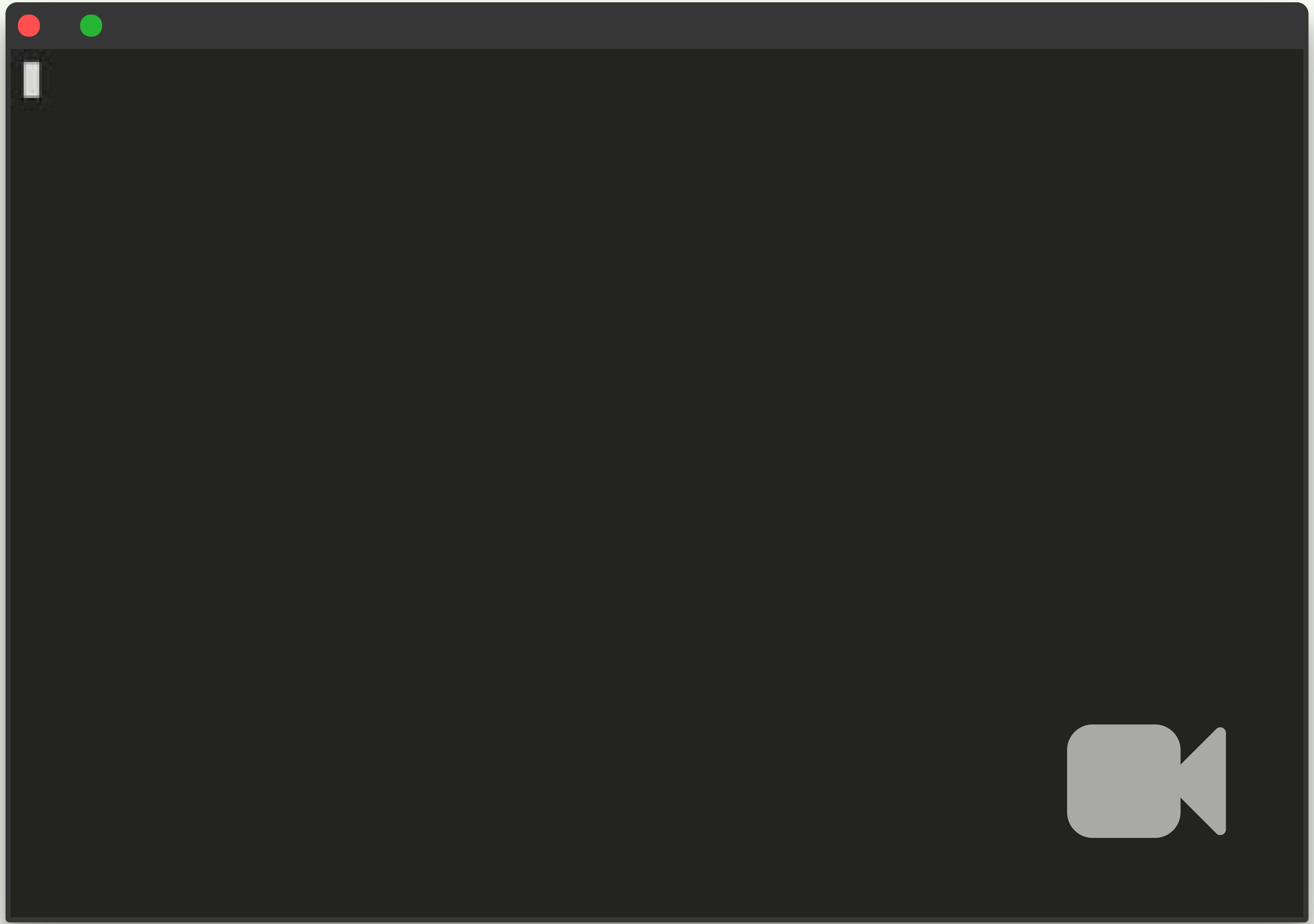
Core File Analysis

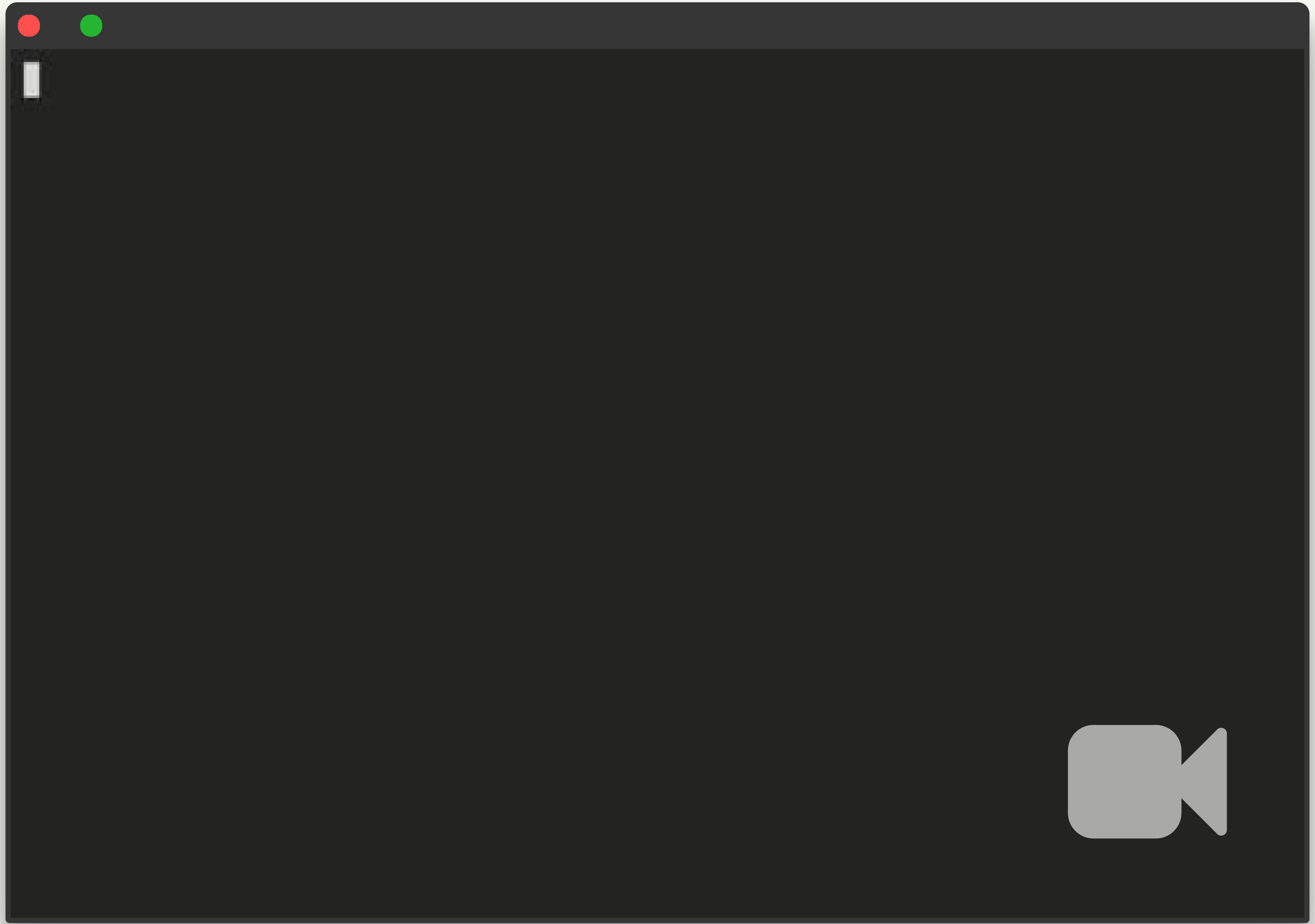
with lldb + llnode

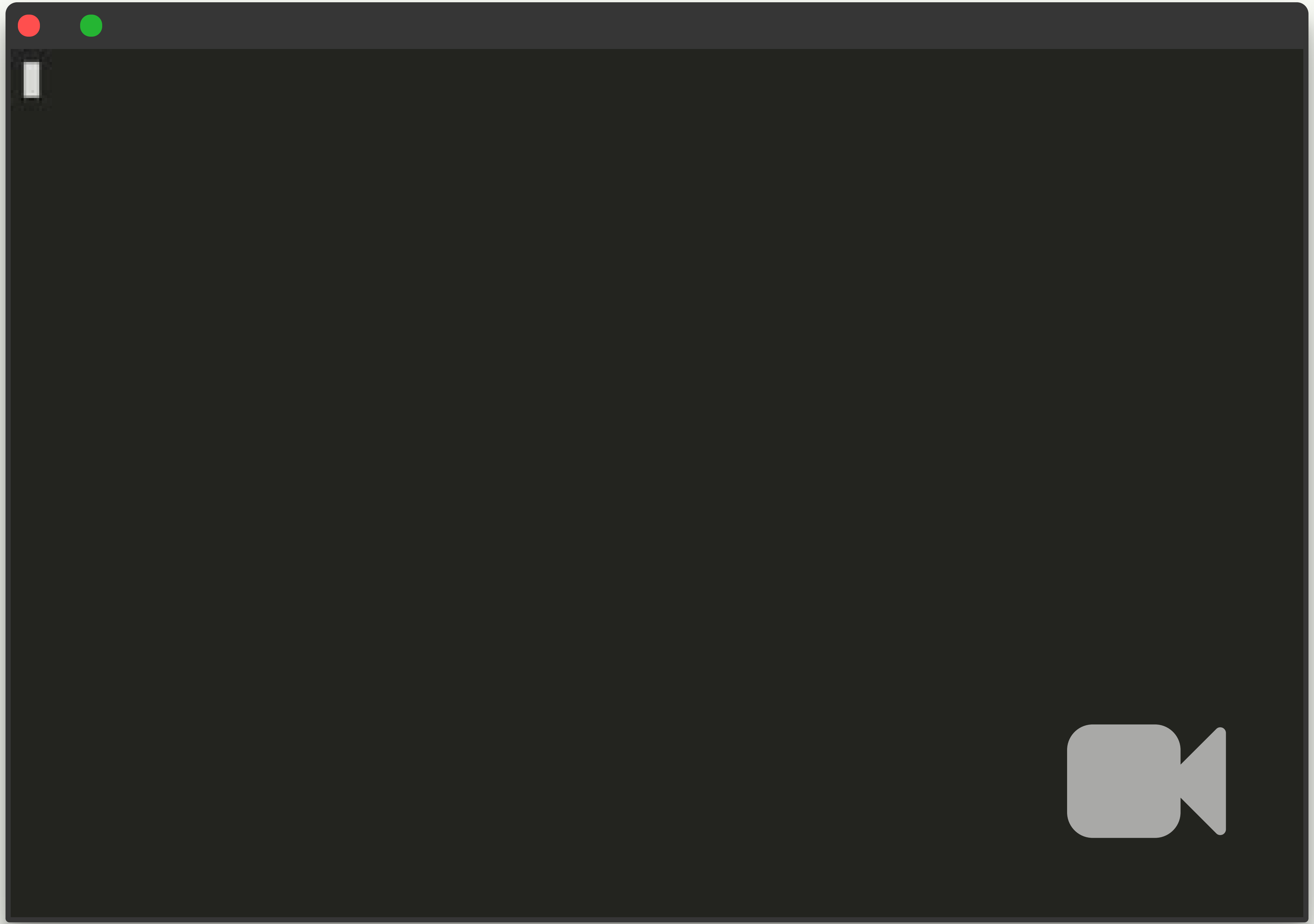


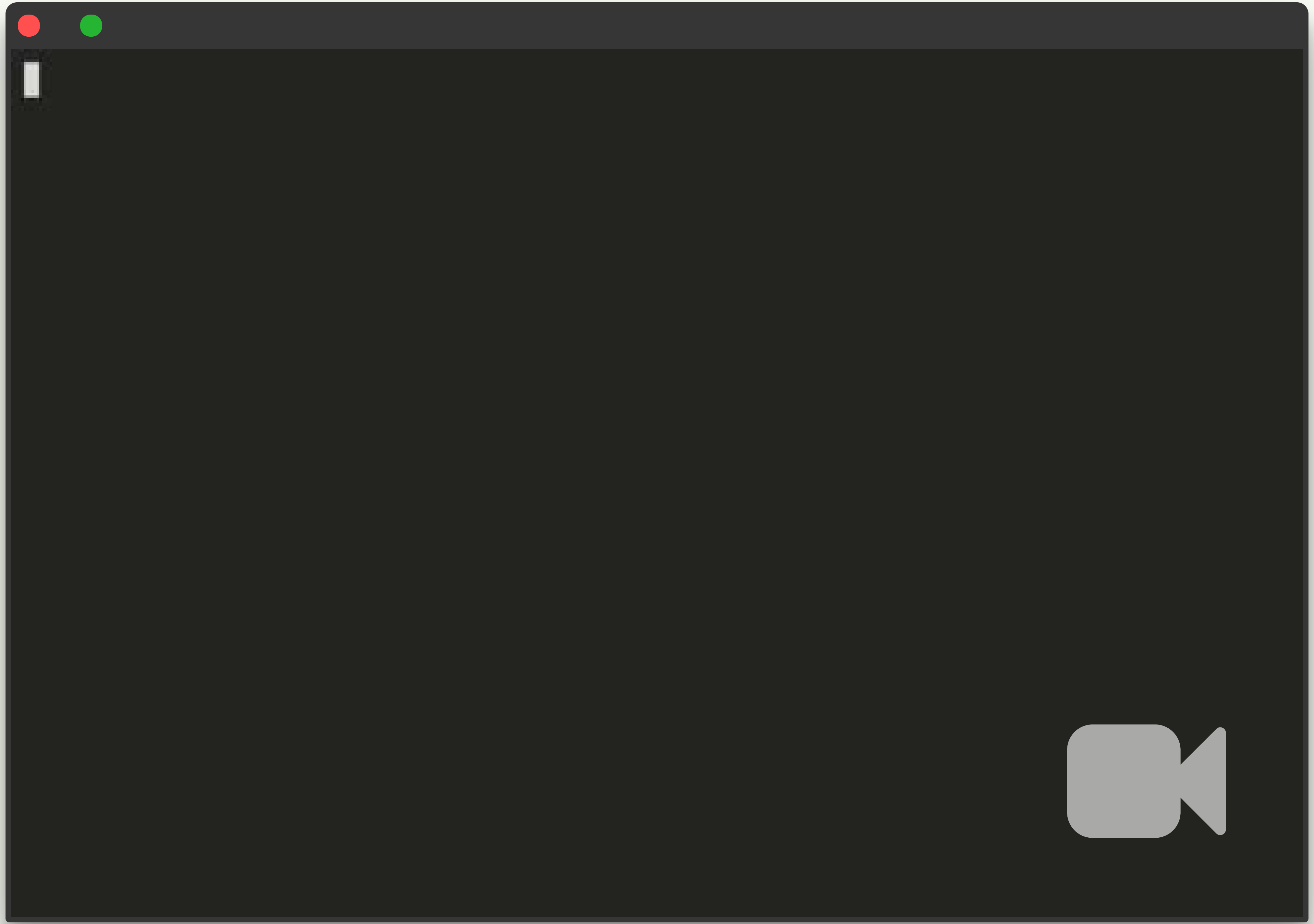
```
const express = require('express')
const problematic = require("./problematic");
const app = express();

app.get('/greatestCommonDivisor/:a/:b/', function (req, res) {
  let a = Number(req.params.a),
      b =  Number(req.params.b);
  let result = problematic.greatestCommonDivisor(a, b);
  let message = "The Greater Common Divisor is: ";
  message = message + result.toString() + "\n";
  res.send(message);
})

app.listen(3000);
```

index.js







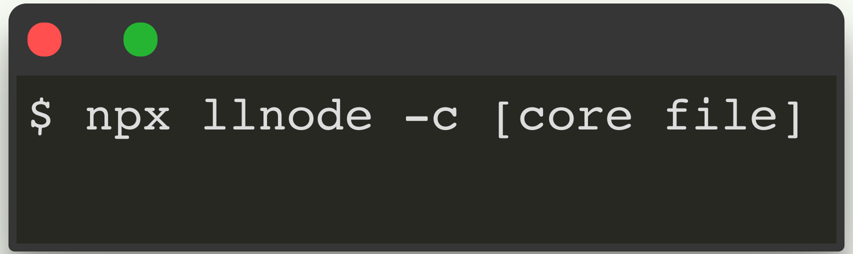


NodeJS

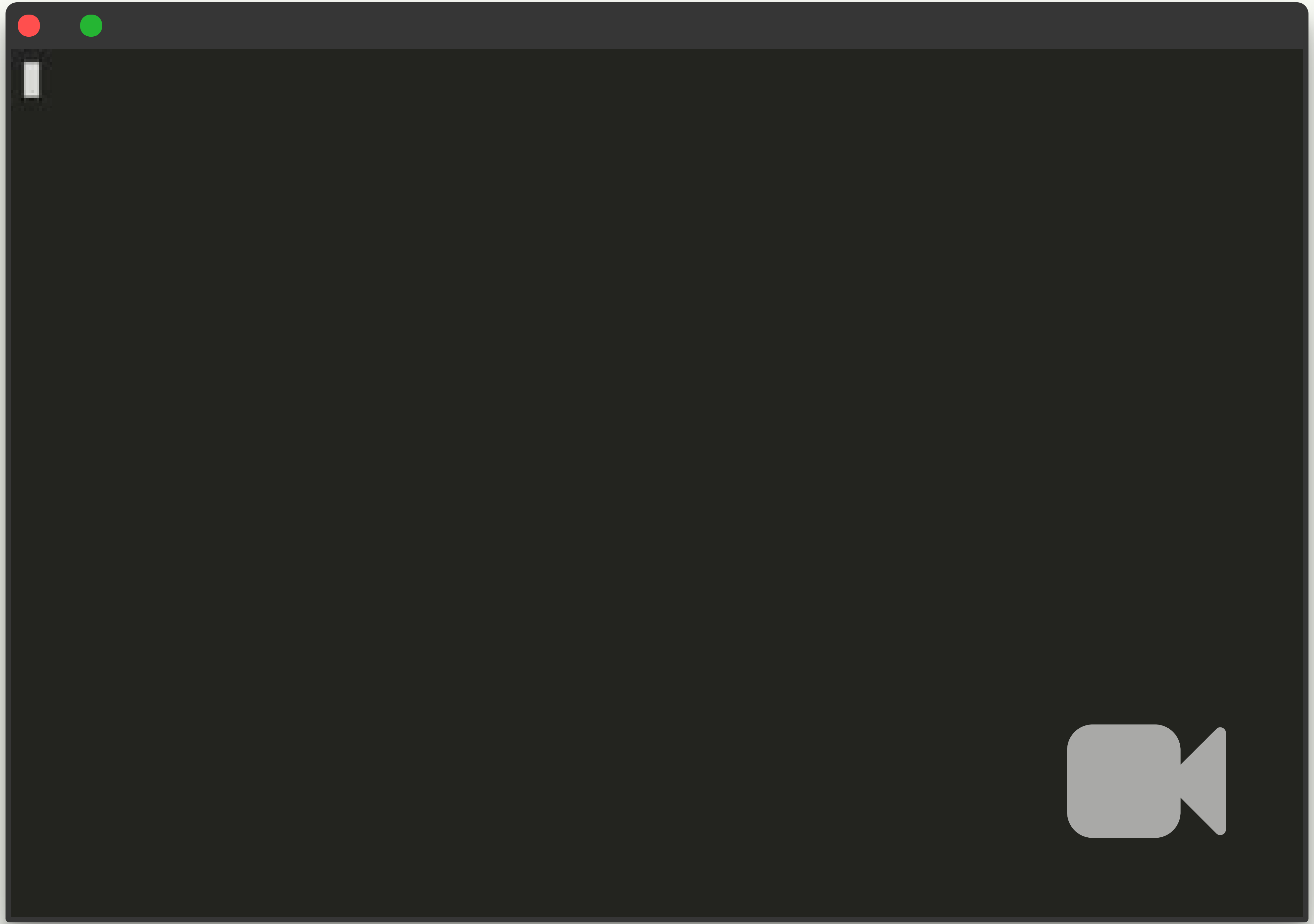
A problem has been detected and NodeJS has been shut down to prevent damage to your computer.

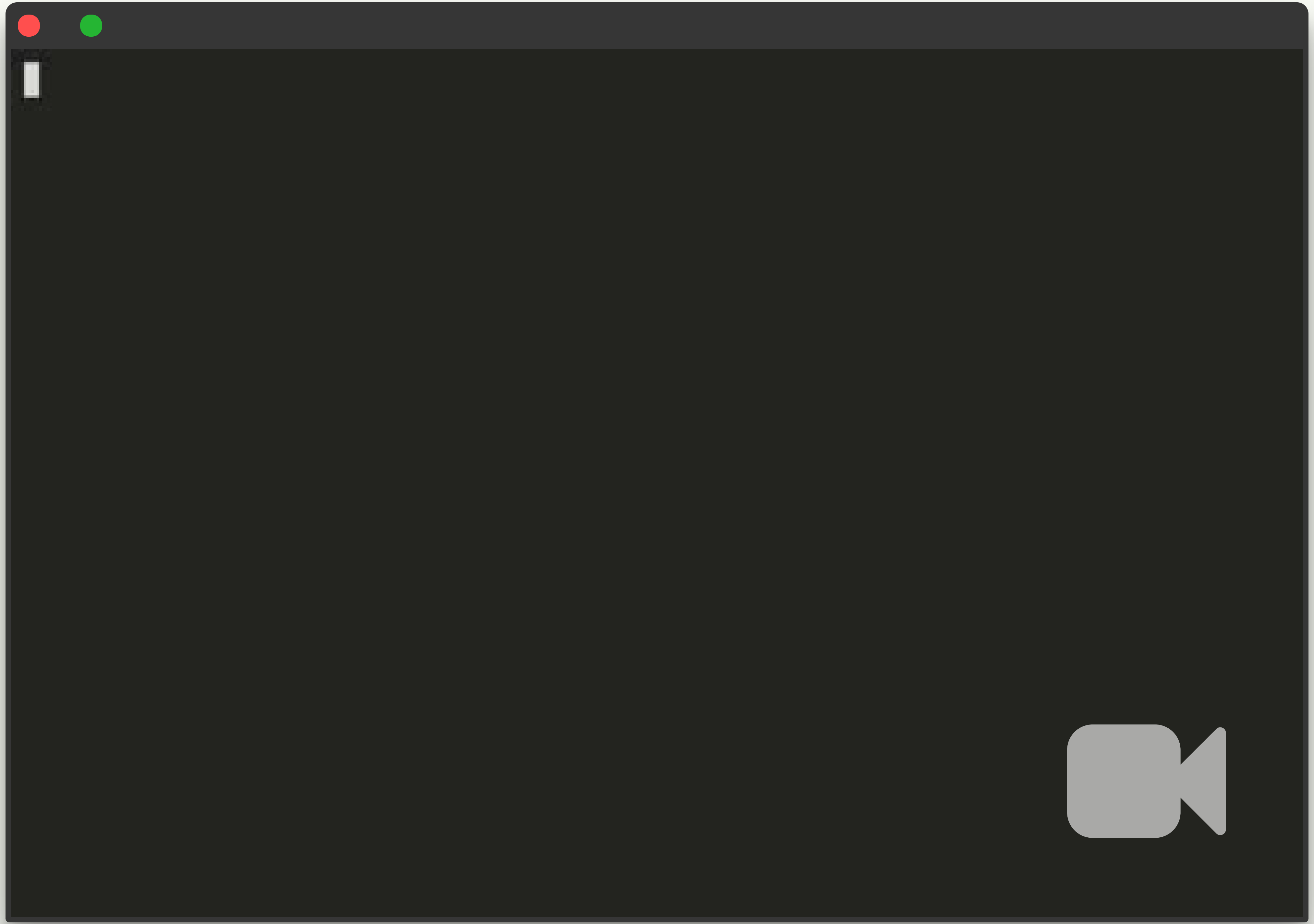
```
<--- Last few GCs --->
[10915:0x3f6b950]     2244 ms: Mark-sweep 30.8 (70.4) -> 30.8 (70.4) MB, 19.2 / 0.0 ms
 allocation failure GC in old space requested
[10915:0x3f6b950]     2269 ms: Mark-sweep 30.8 (70.4) -> 30.8 (39.4) MB, 25.5 / 0.0 ms
 last resort
[10915:0x3f6b950]     2294 ms: Mark-sweep 30.8 (39.4) -> 30.8 (39.4) MB, 24.3 / 0.0 ms
 last resort

FATAL ERROR: CALL_AND_RETRY_LAST Allocation failed - JavaScript heap out of memory

Aborted (core dumped)
```

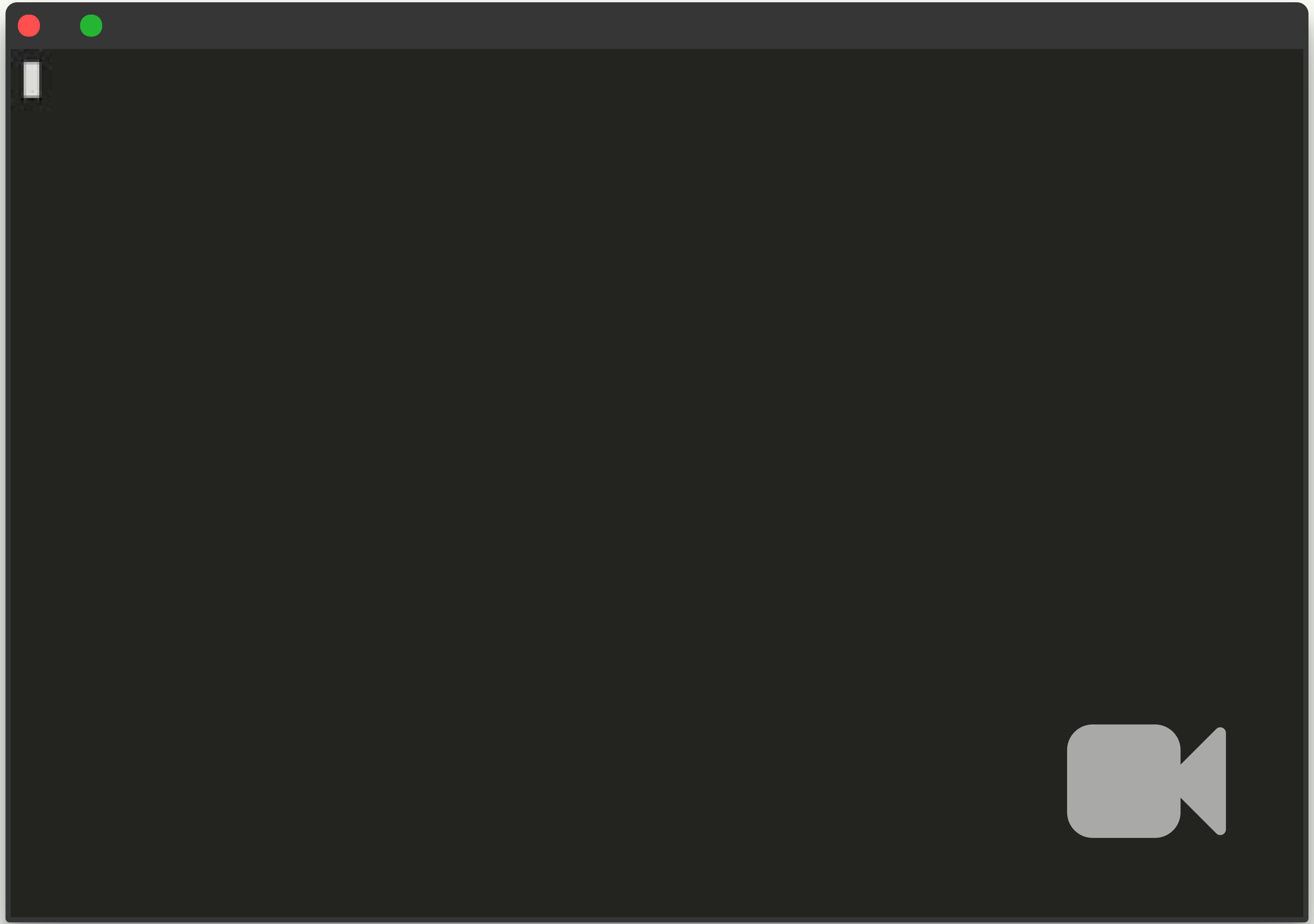
Press any key to continue _

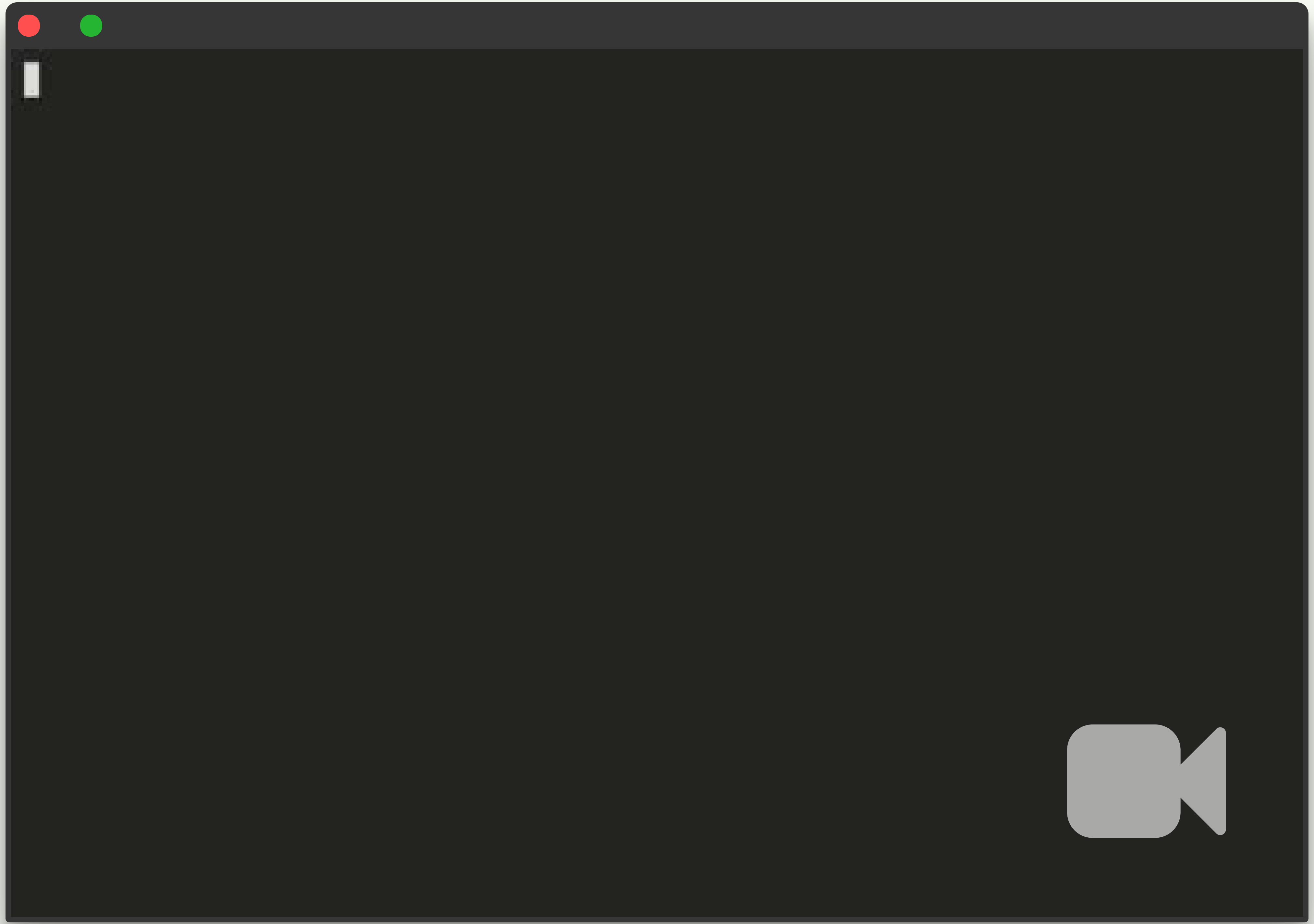


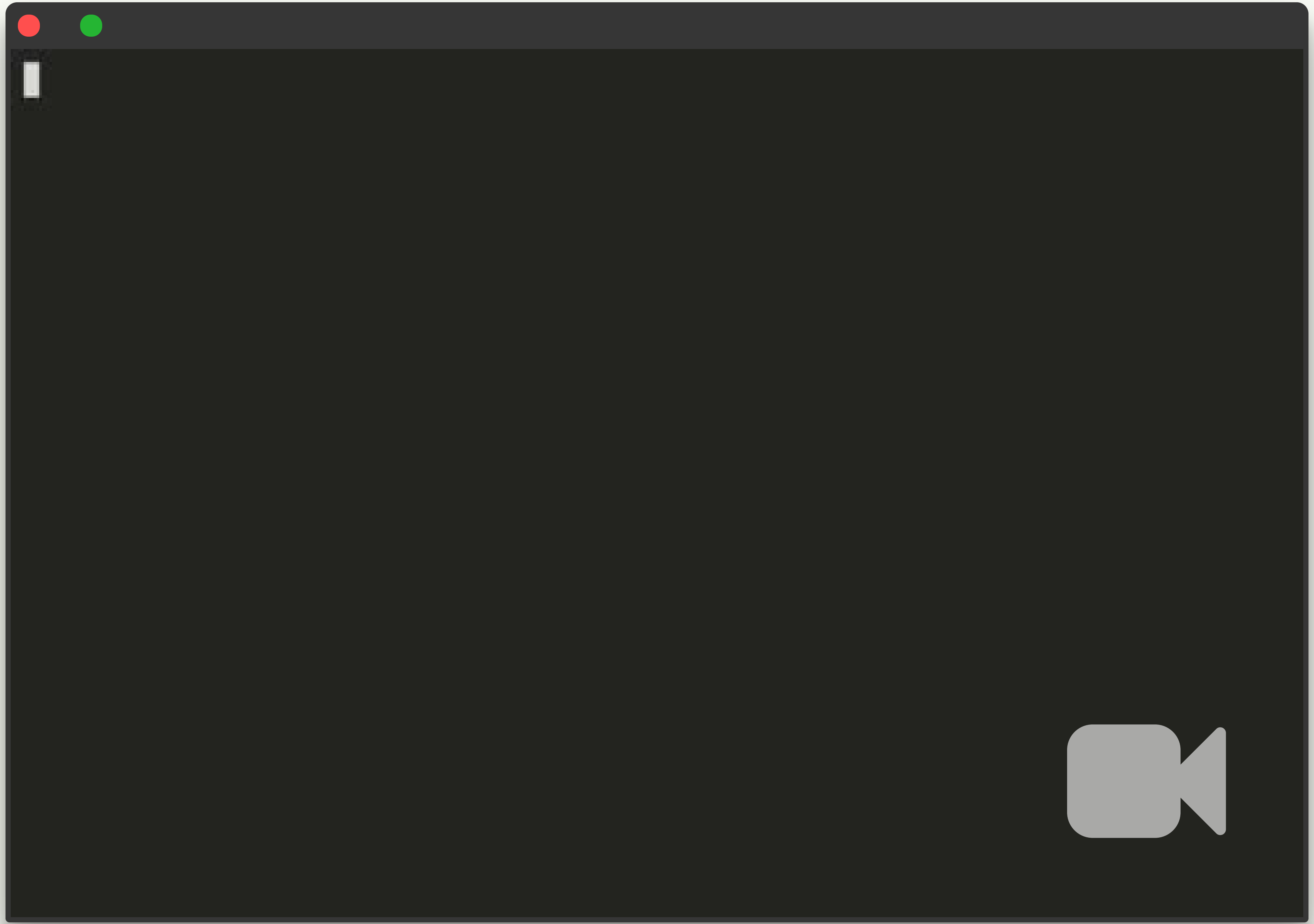
```
$ npx llnode -c [core file]
```

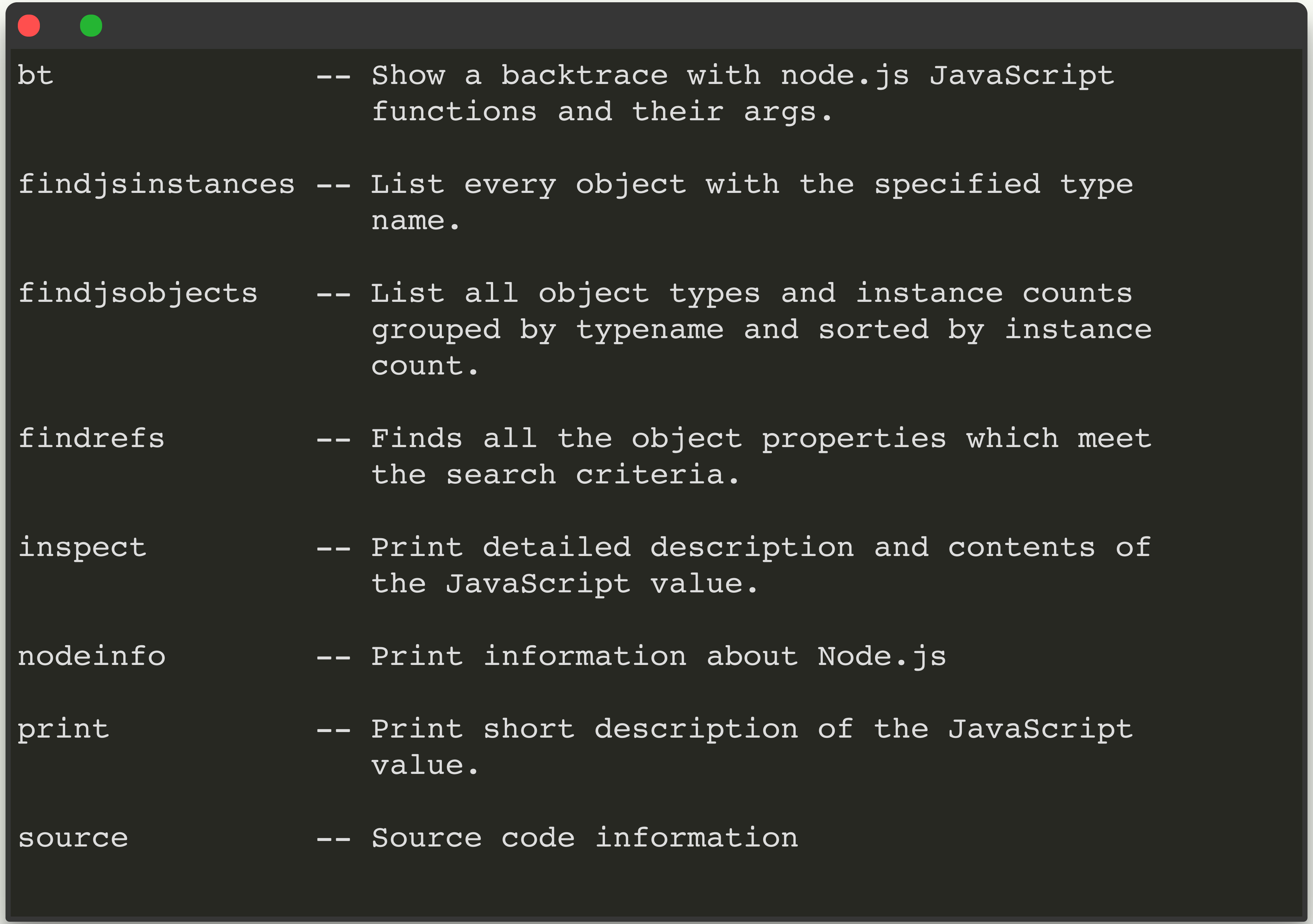




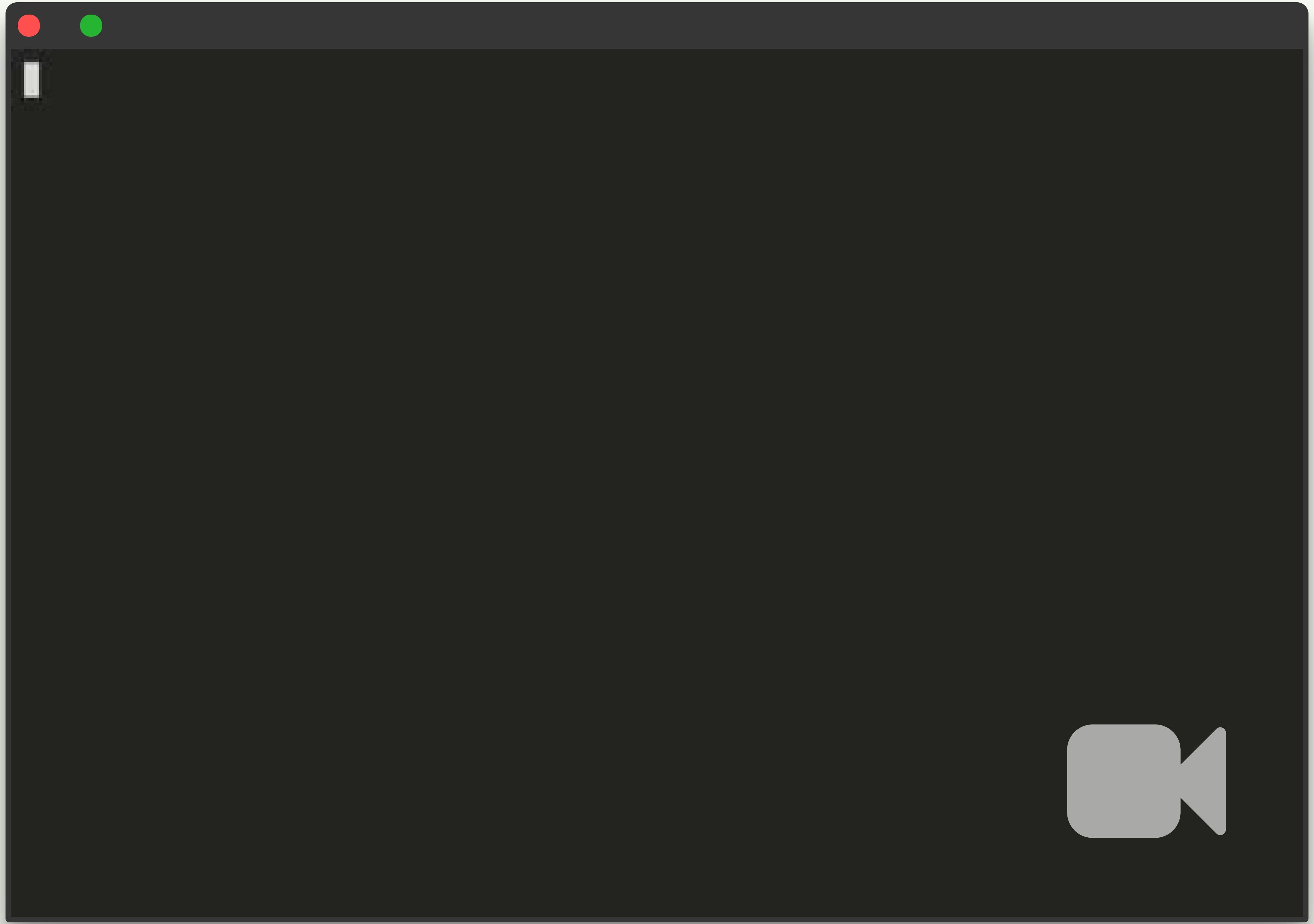


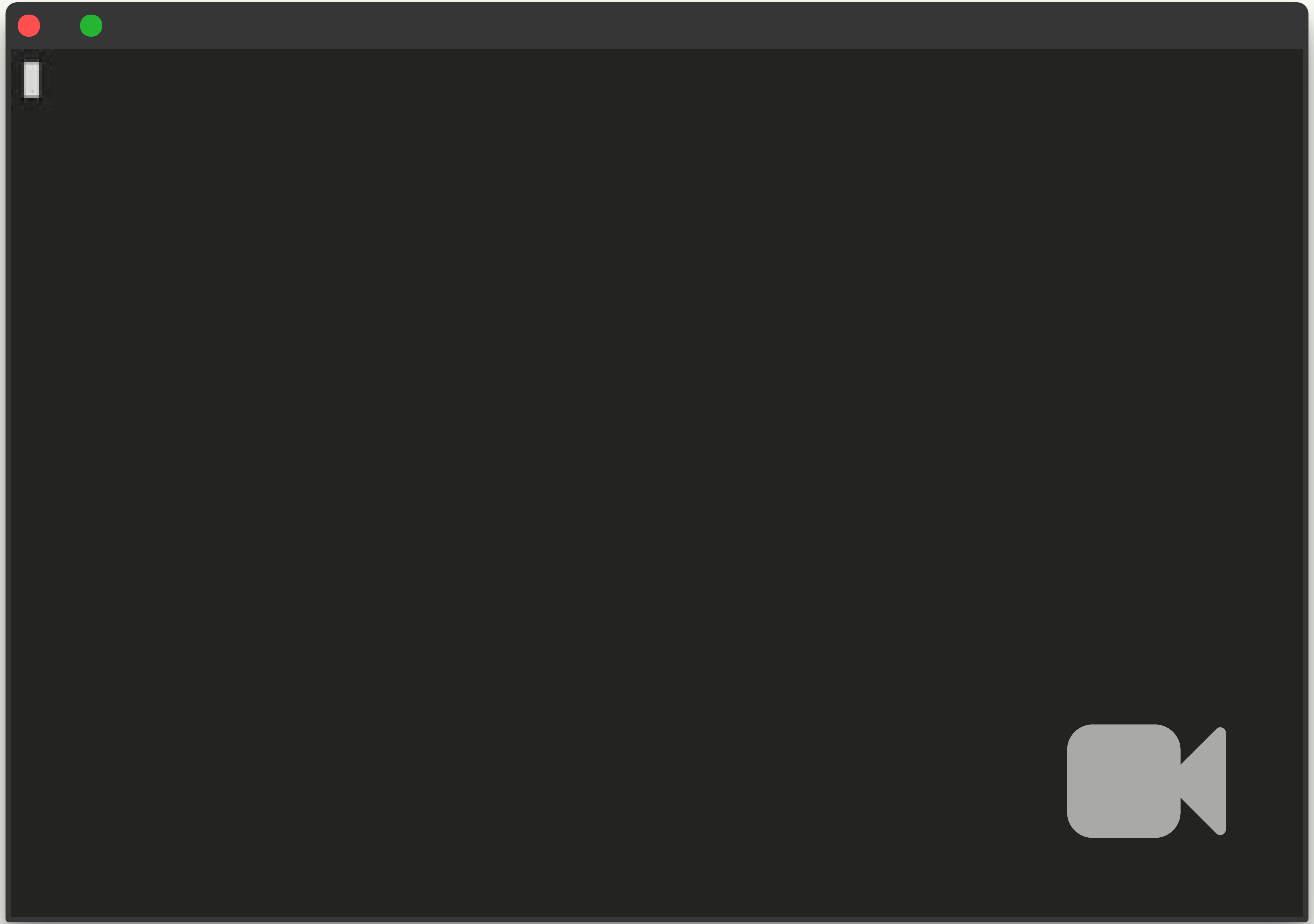


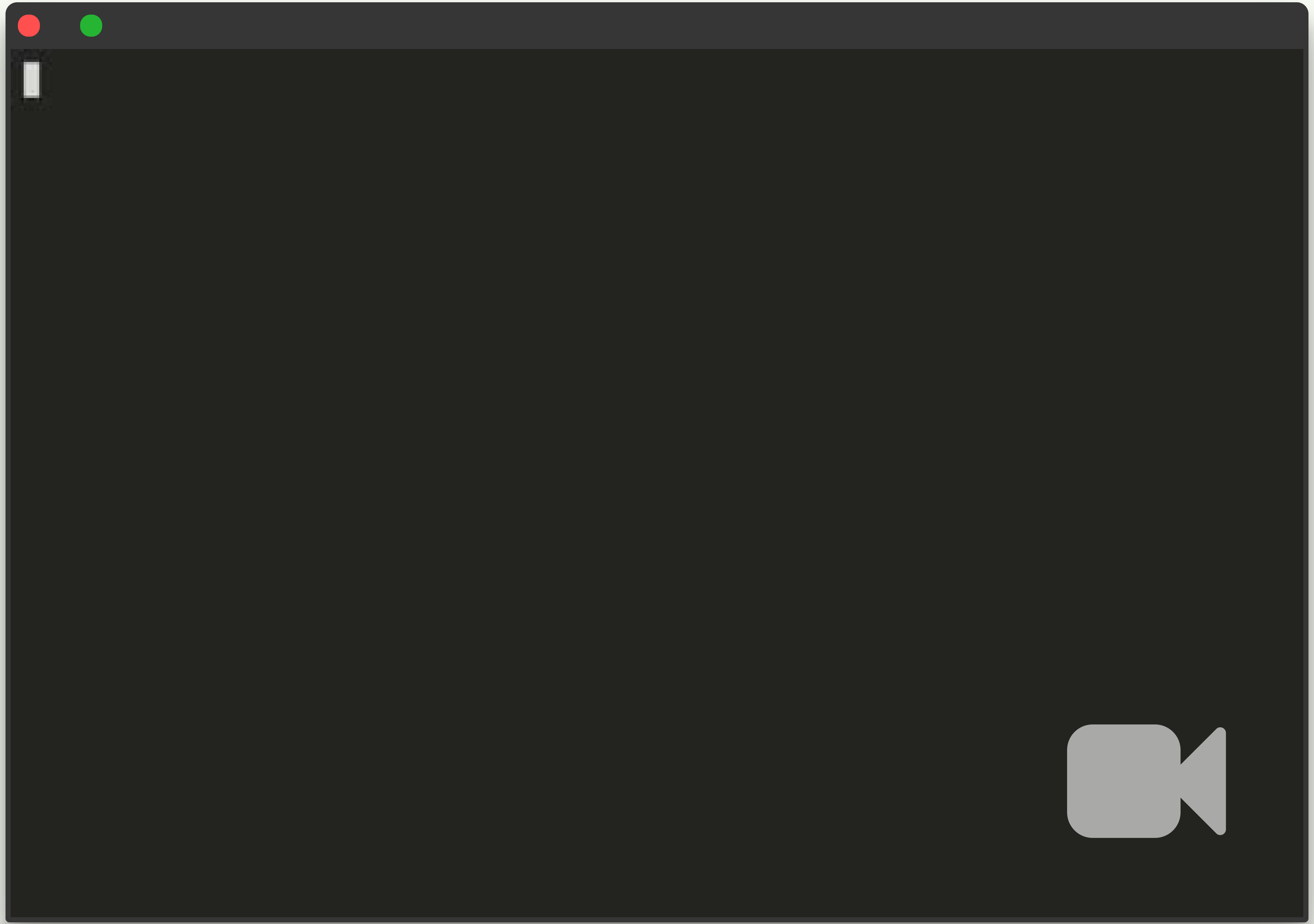


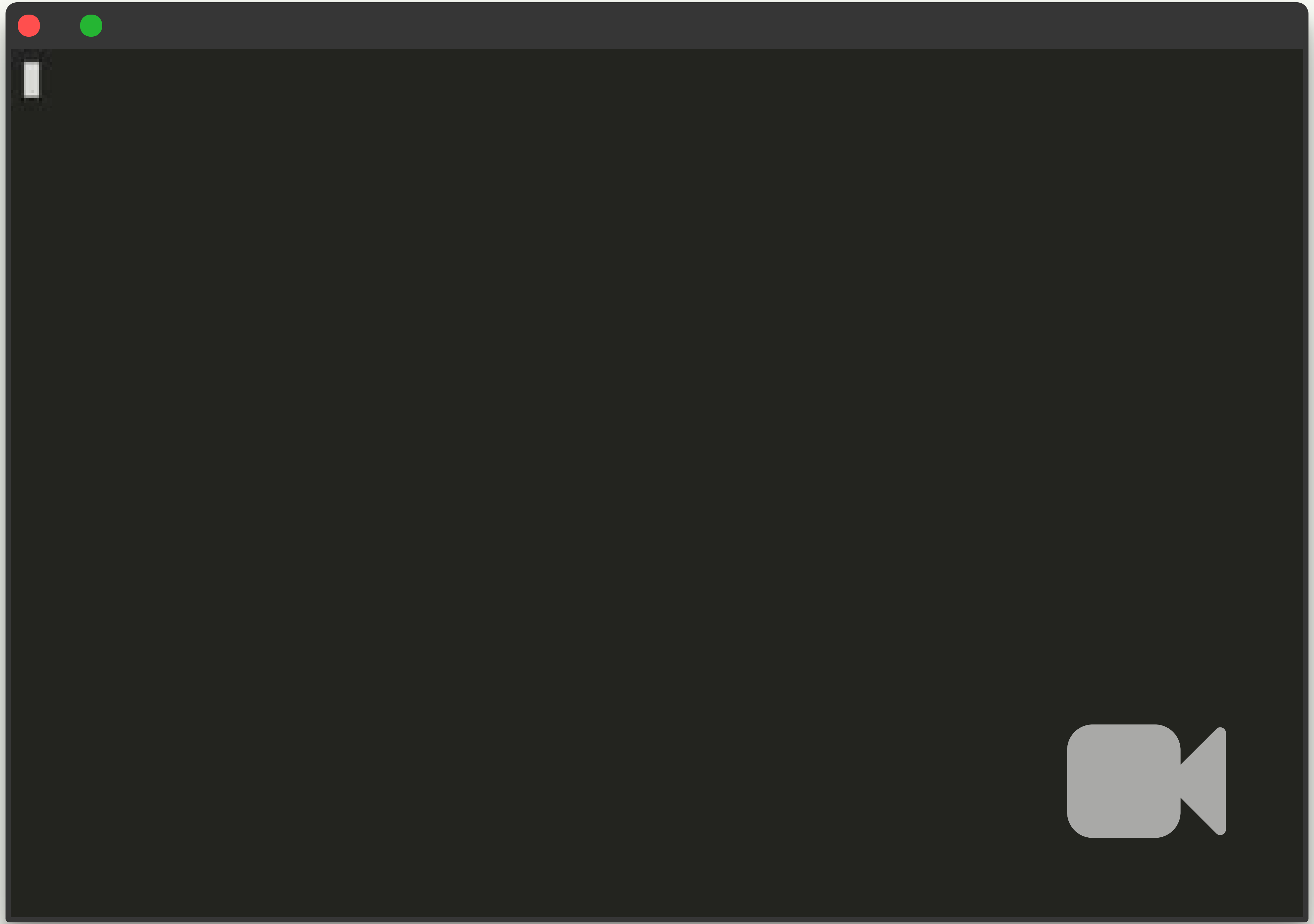


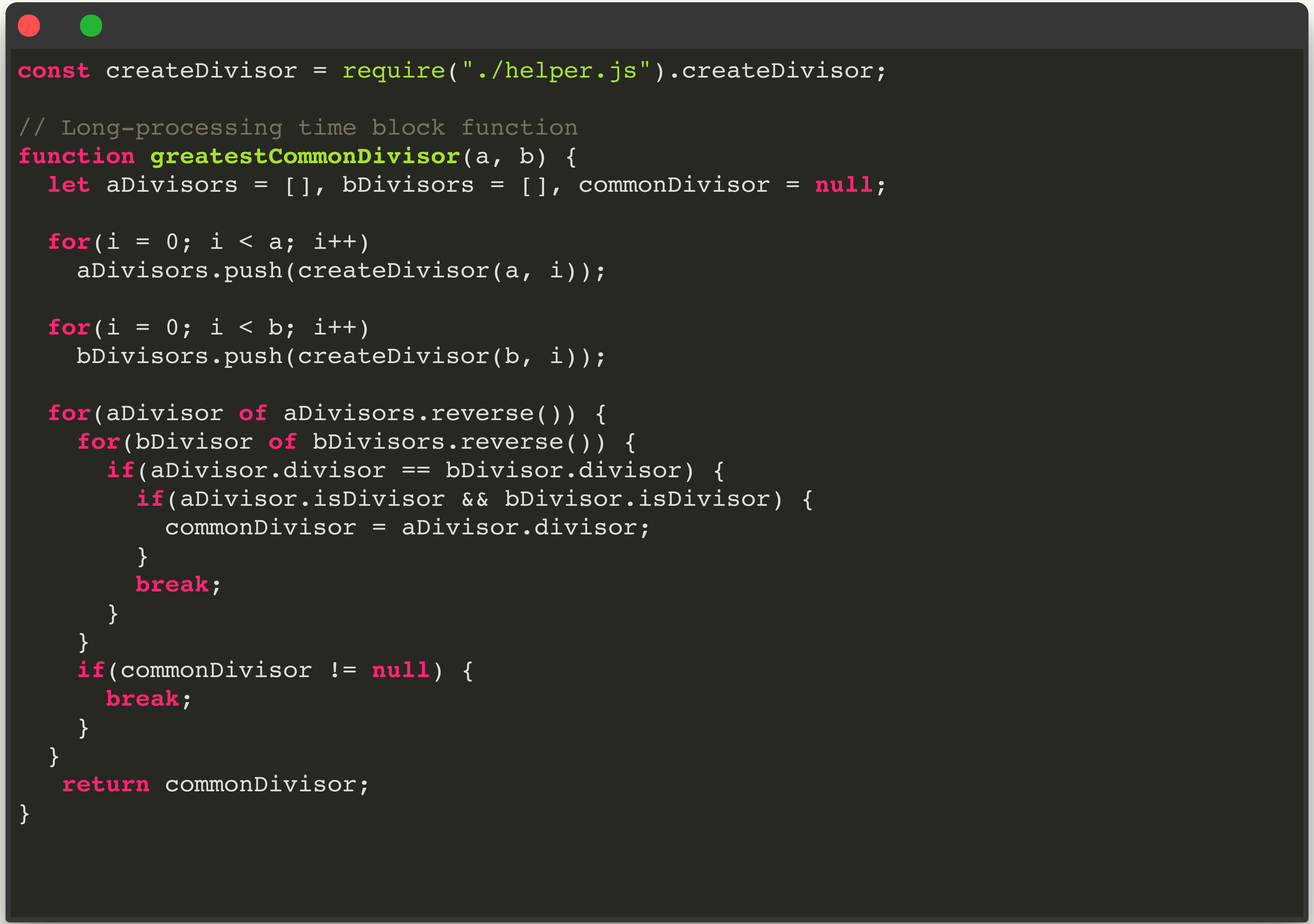
```
bt             -- Show a backtrace with node.js JavaScript
                  functions and their args.

findjsinstances -- List every object with the specified type
                  name.

findjsobjects  -- List all object types and instance counts
                  grouped by typename and sorted by instance
                  count.

findrefs       -- Finds all the object properties which meet
                  the search criteria.

inspect        -- Print detailed description and contents of
                  the JavaScript value.

nodeinfo       -- Print information about Node.js

print          -- Print short description of the JavaScript
                  value.

source         -- Source code information
```





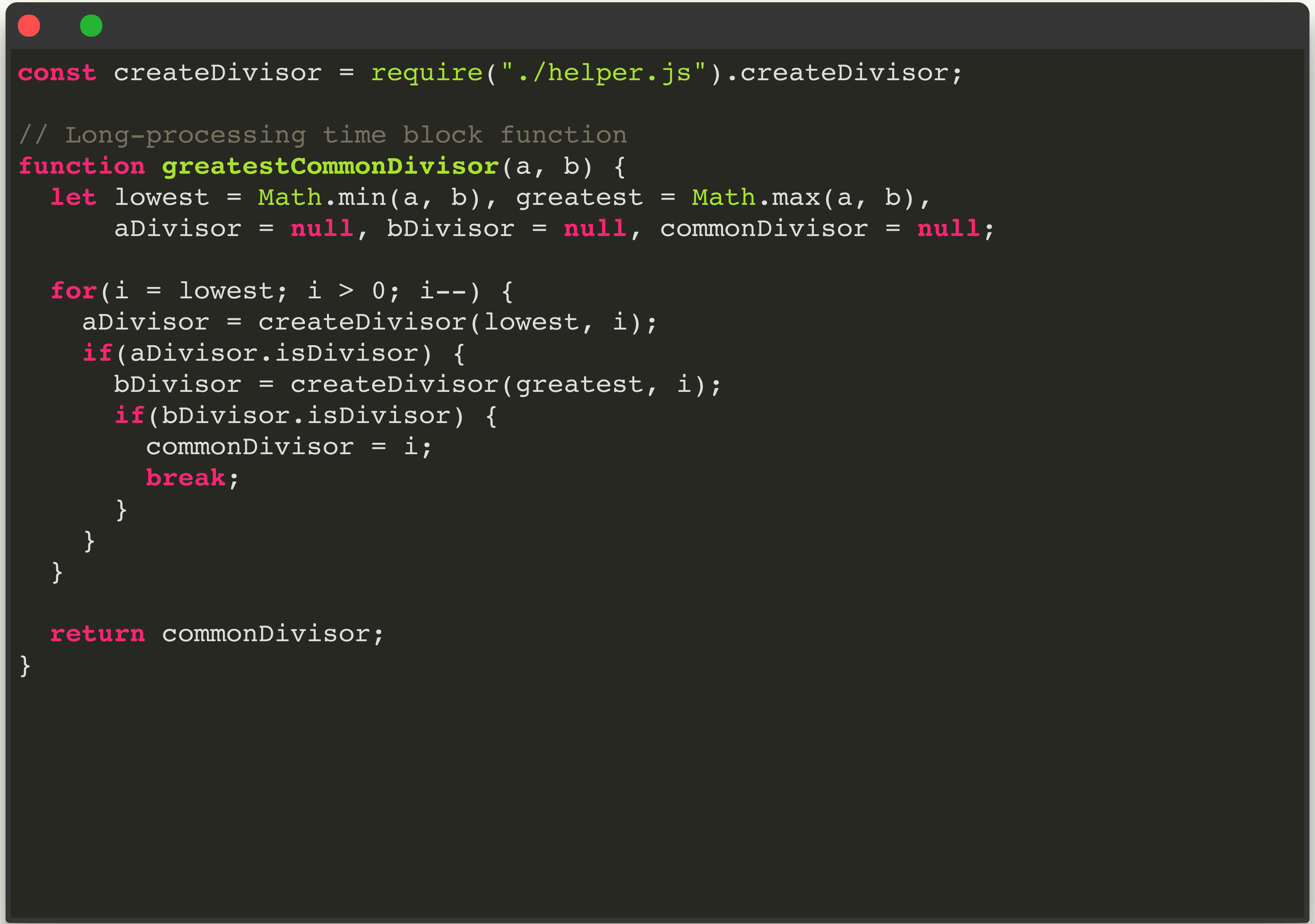






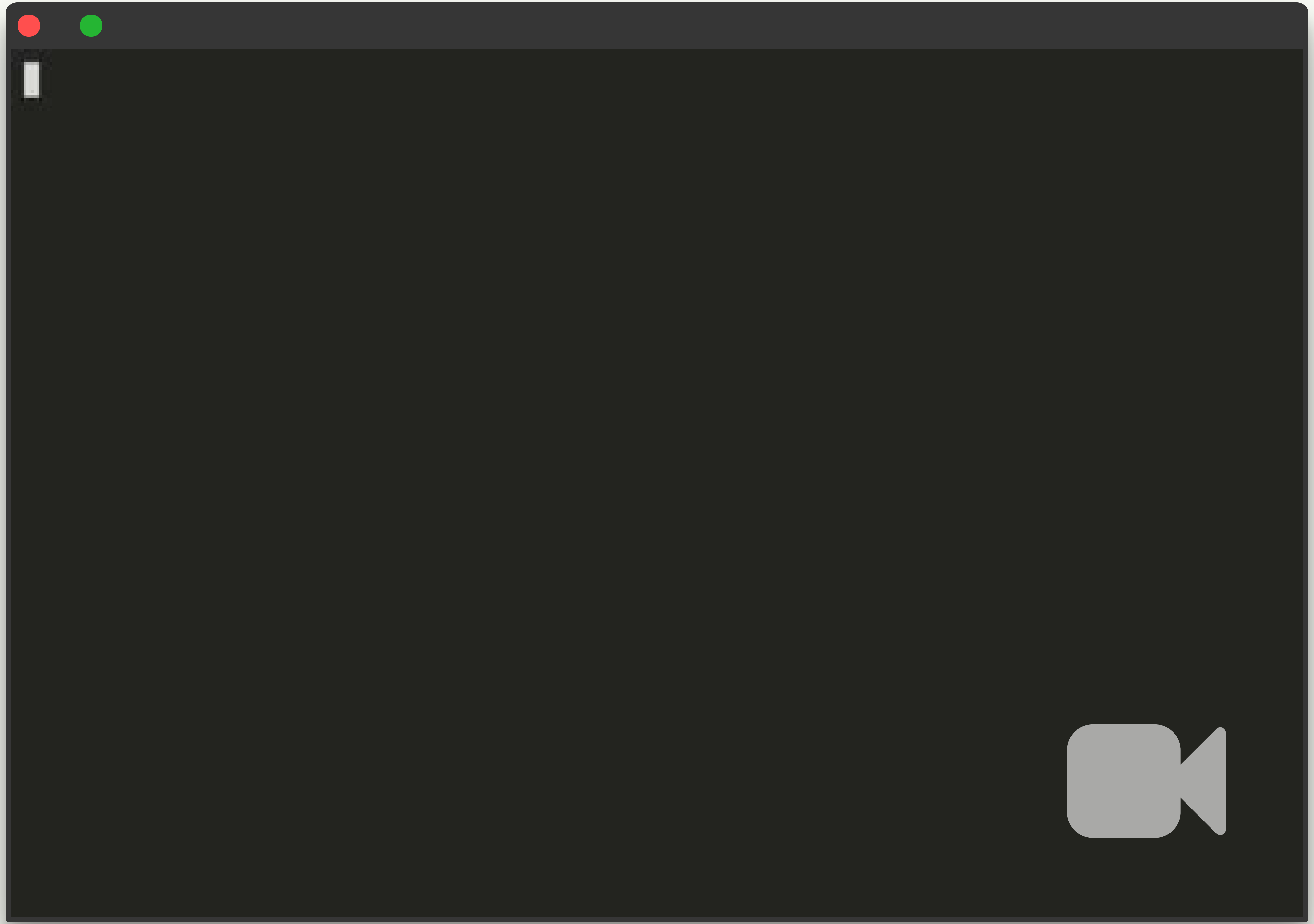
```
const createDivisor = require("./helper.js").createDivisor;

// Long-processing time block function
function greatestCommonDivisor(a, b) {
  let aDivisors = [], bDivisors = [], commonDivisor = null;

  for(i = 0; i < a; i++)
    aDivisors.push(createDivisor(a, i));

  for(i = 0; i < b; i++)
    bDivisors.push(createDivisor(b, i));

  for(aDivisor of aDivisors.reverse()) {
    for(bDivisor of bDivisors.reverse()) {
      if(aDivisor.divisor == bDivisor.divisor) {
        if(aDivisor.isDivisor && bDivisor.isDivisor) {
          commonDivisor = aDivisor.divisor;
        }
        break;
      }
    }
    if(commonDivisor != null) {
      break;
    }
  }
   return commonDivisor;
}
```

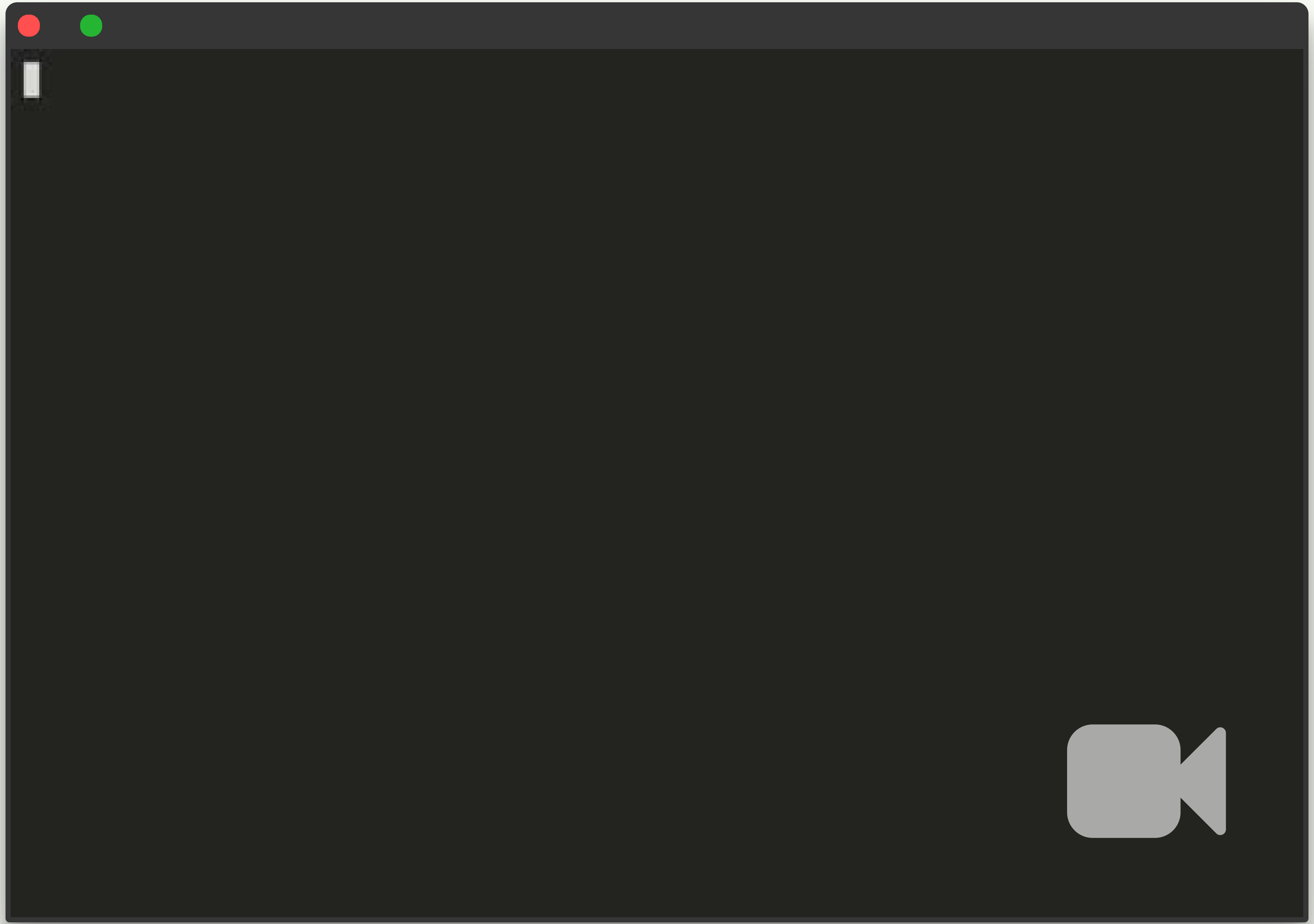
problematic/index.js

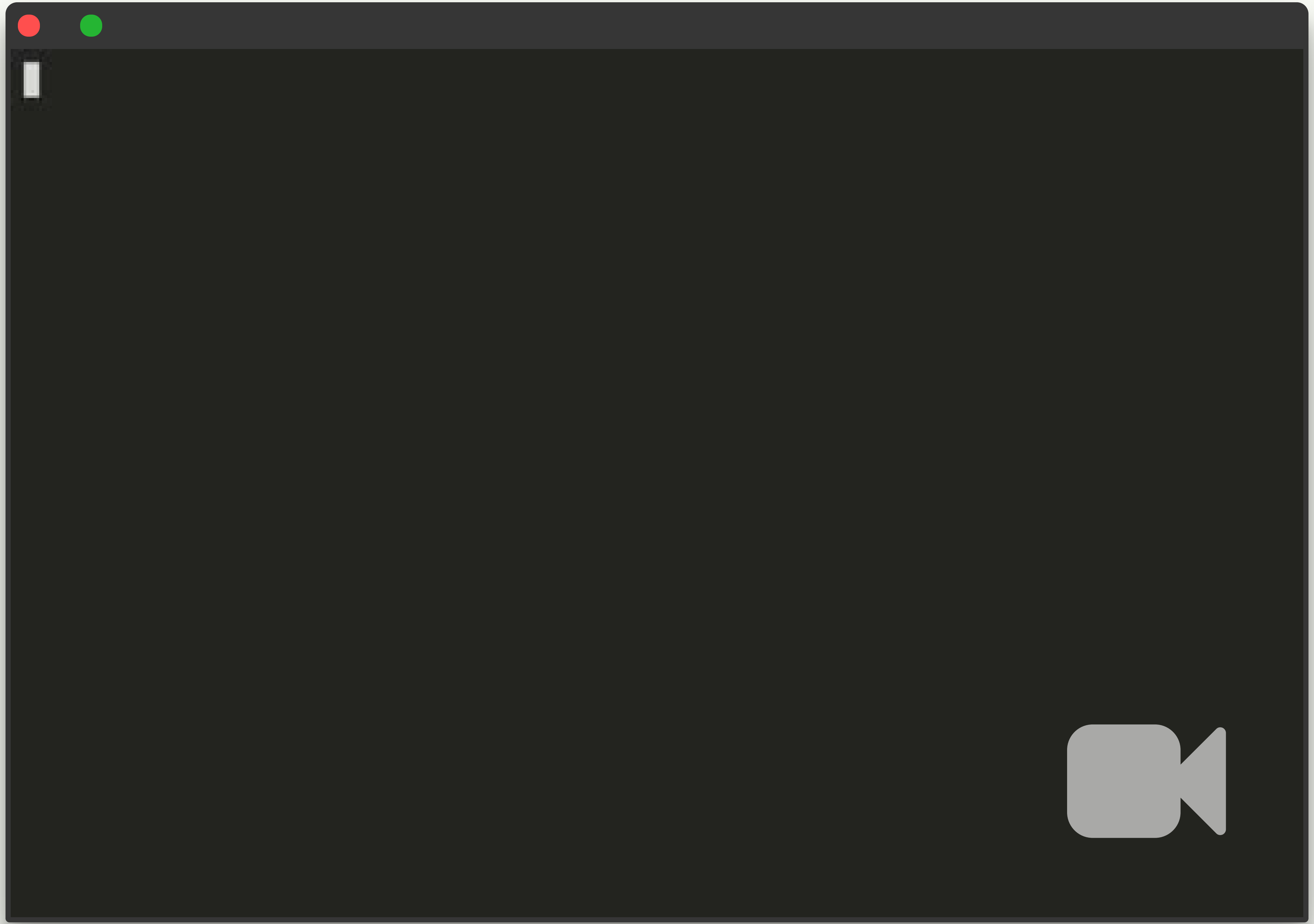


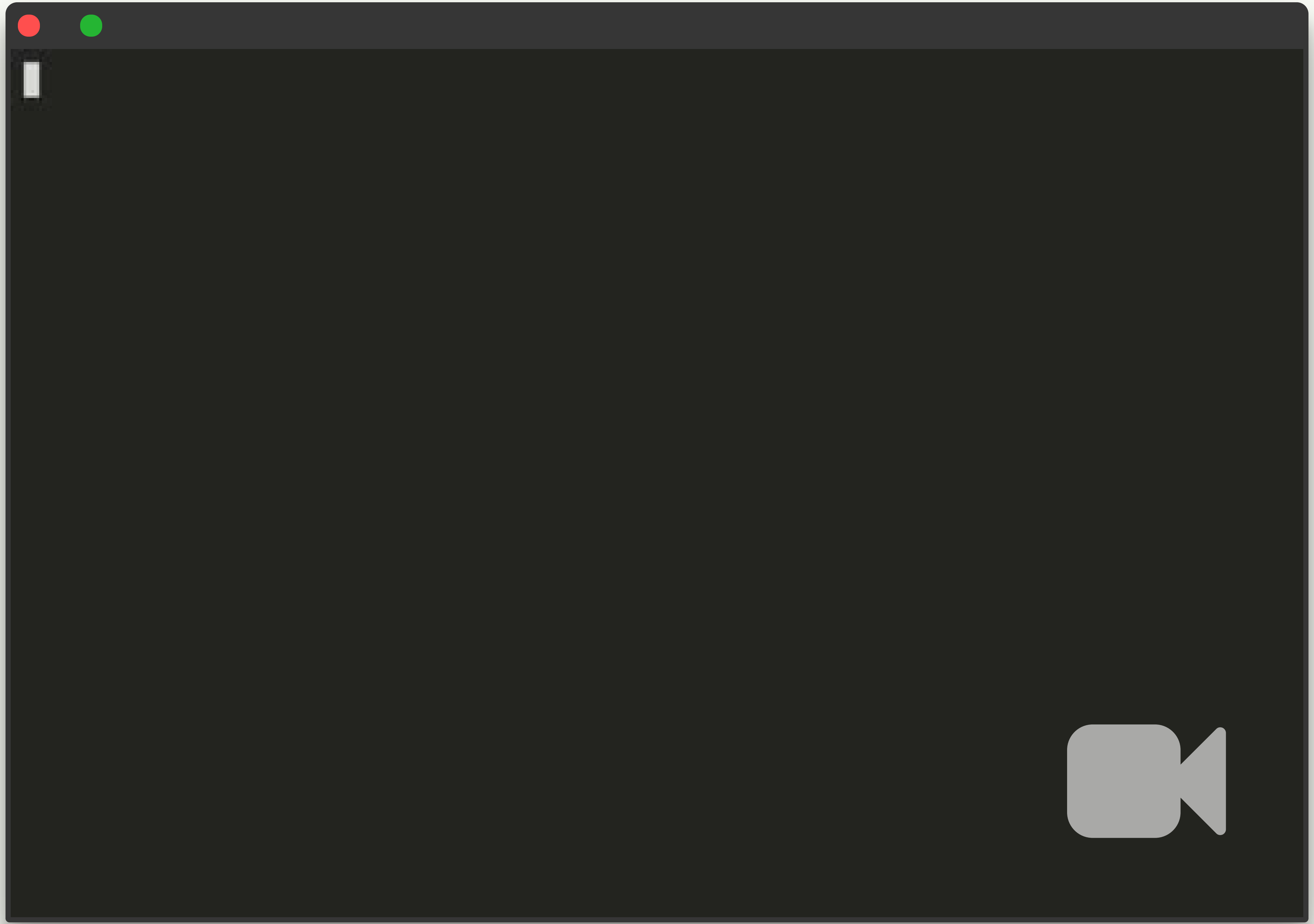
```
const createDivisor = require("./helper.js").createDivisor;

// Long-processing time block function
function greatestCommonDivisor(a, b) {
  let lowest = Math.min(a, b), greatest = Math.max(a, b),
      aDivisor = null, bDivisor = null, commonDivisor = null;

  for(i = lowest; i > 0; i--) {
    aDivisor = createDivisor(lowest, i);
    if(aDivisor.isDivisor) {
      bDivisor = createDivisor(greatest, i);
      if(bDivisor.isDivisor) {
        commonDivisor = i;
        break;
      }
    }
  }

  return commonDivisor;
}
```

problematic/index.js *







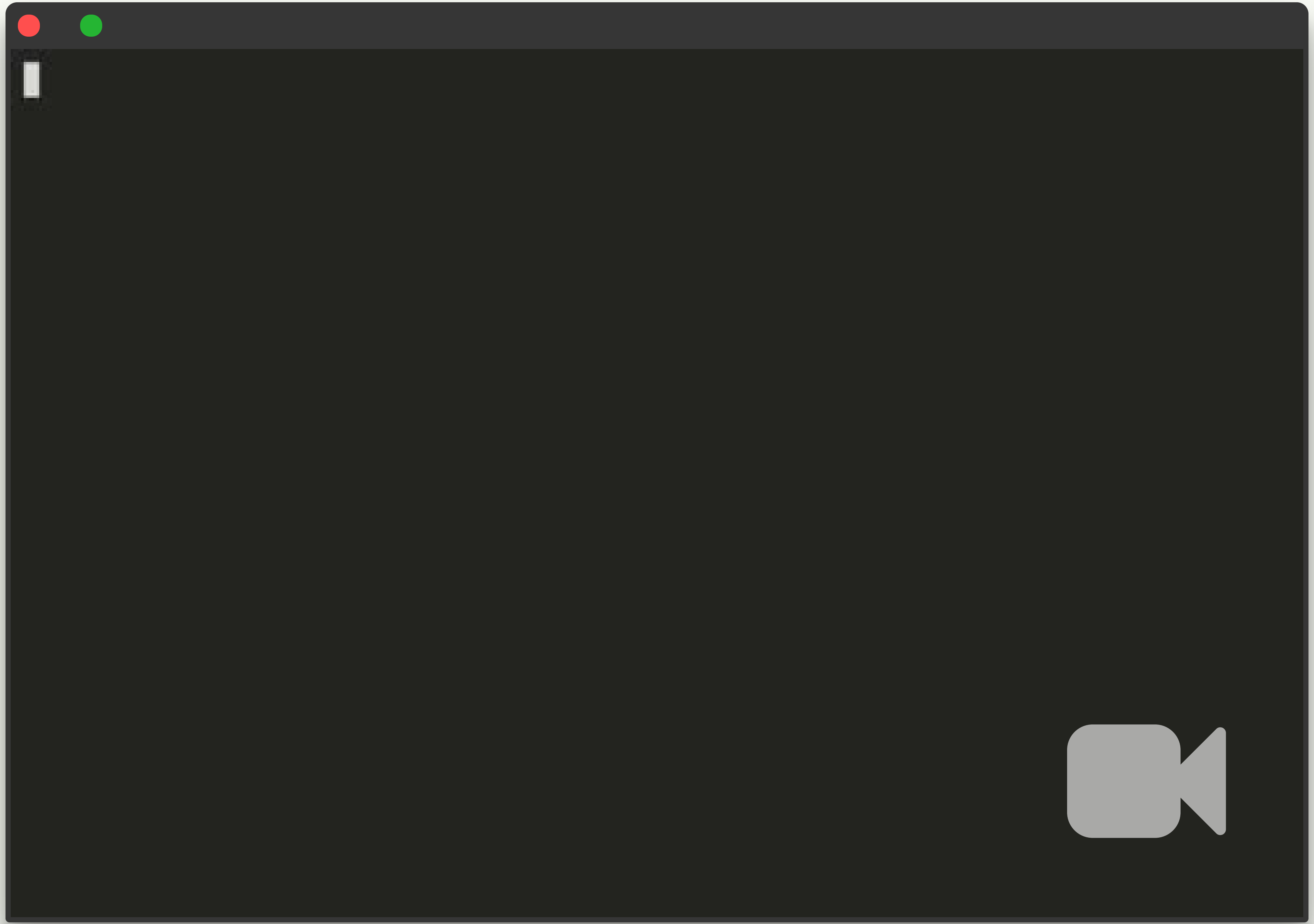


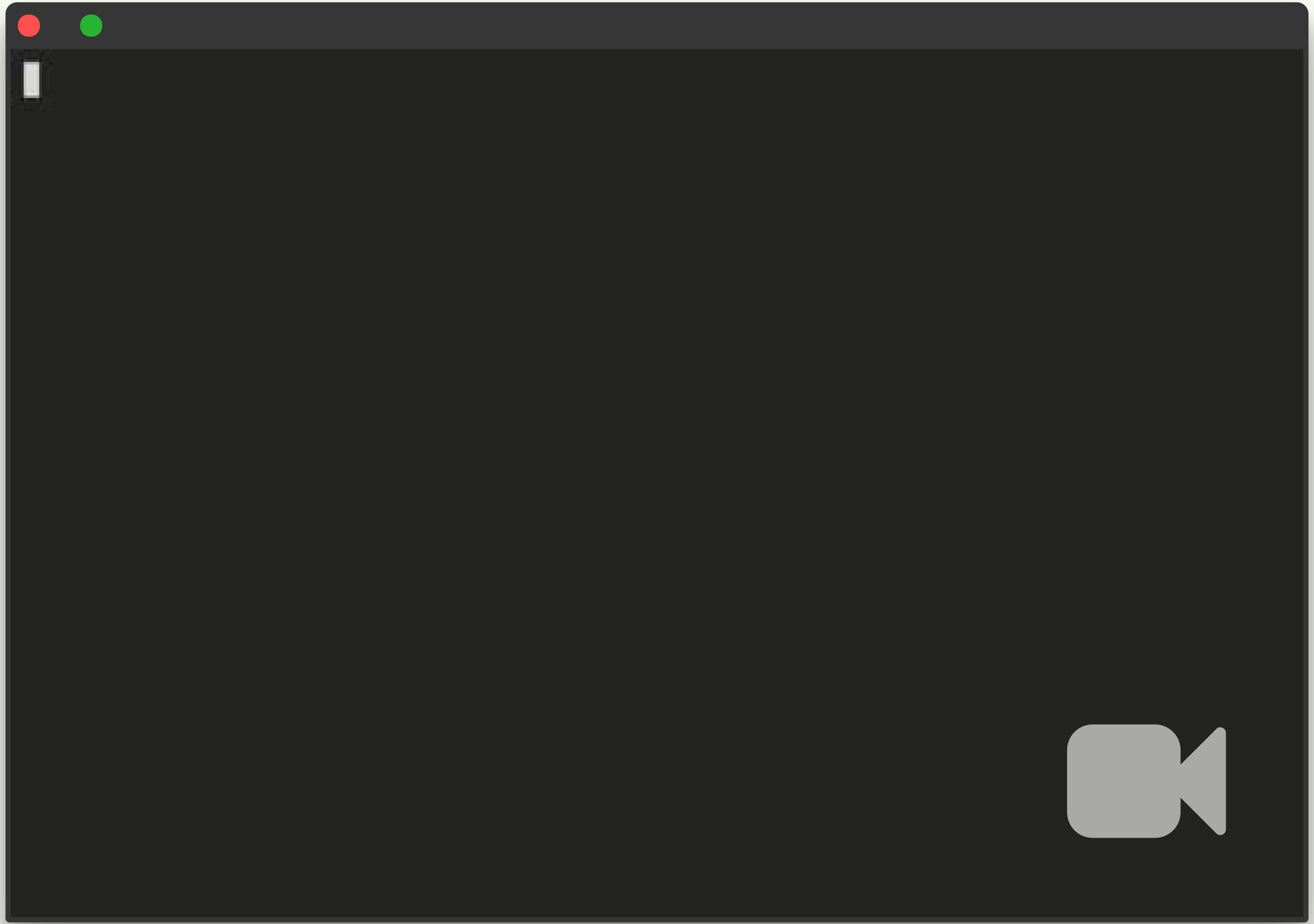
NodeJS

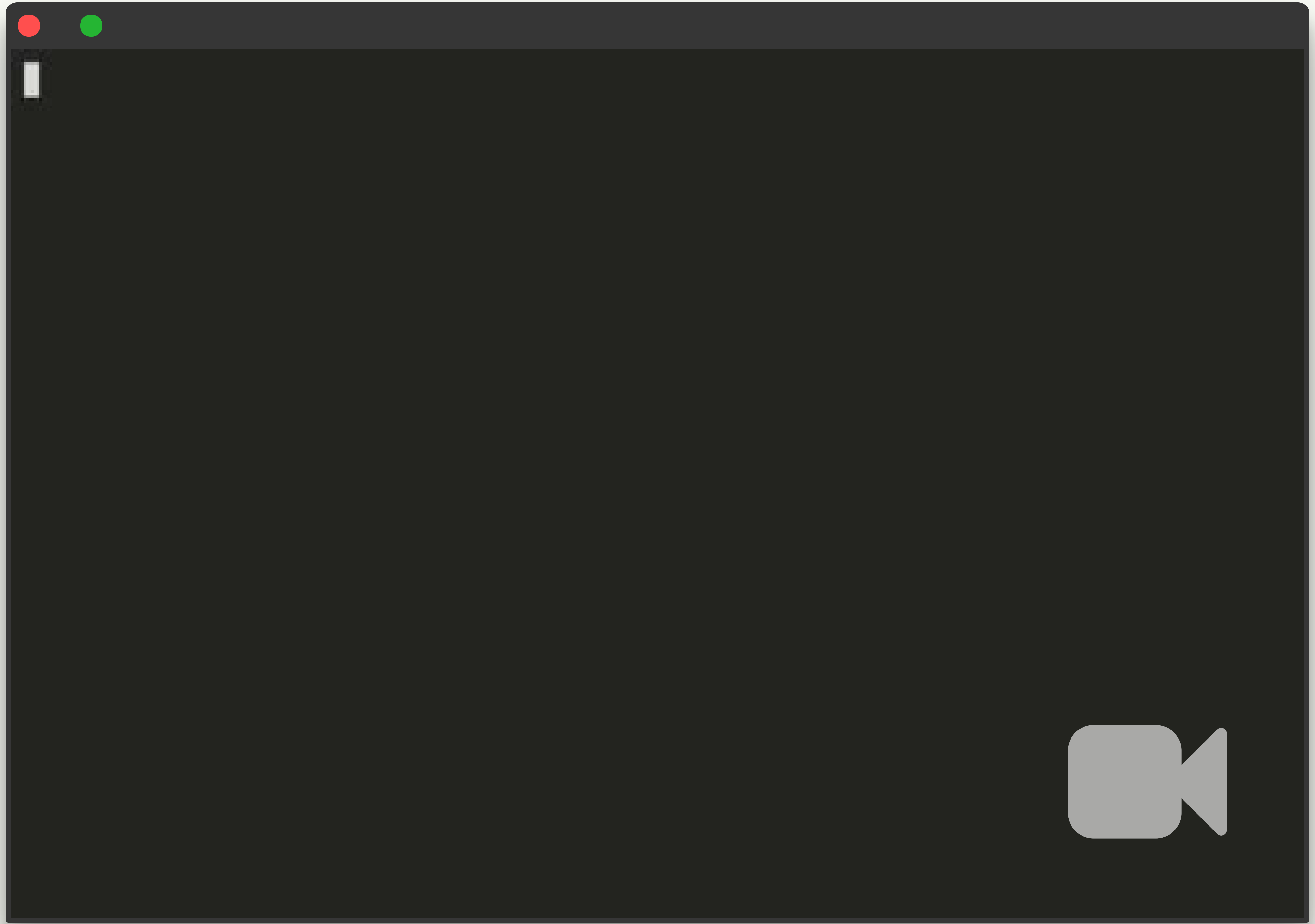
A problem has been detected and NodeJS has been shut down to prevent damage to your computer.

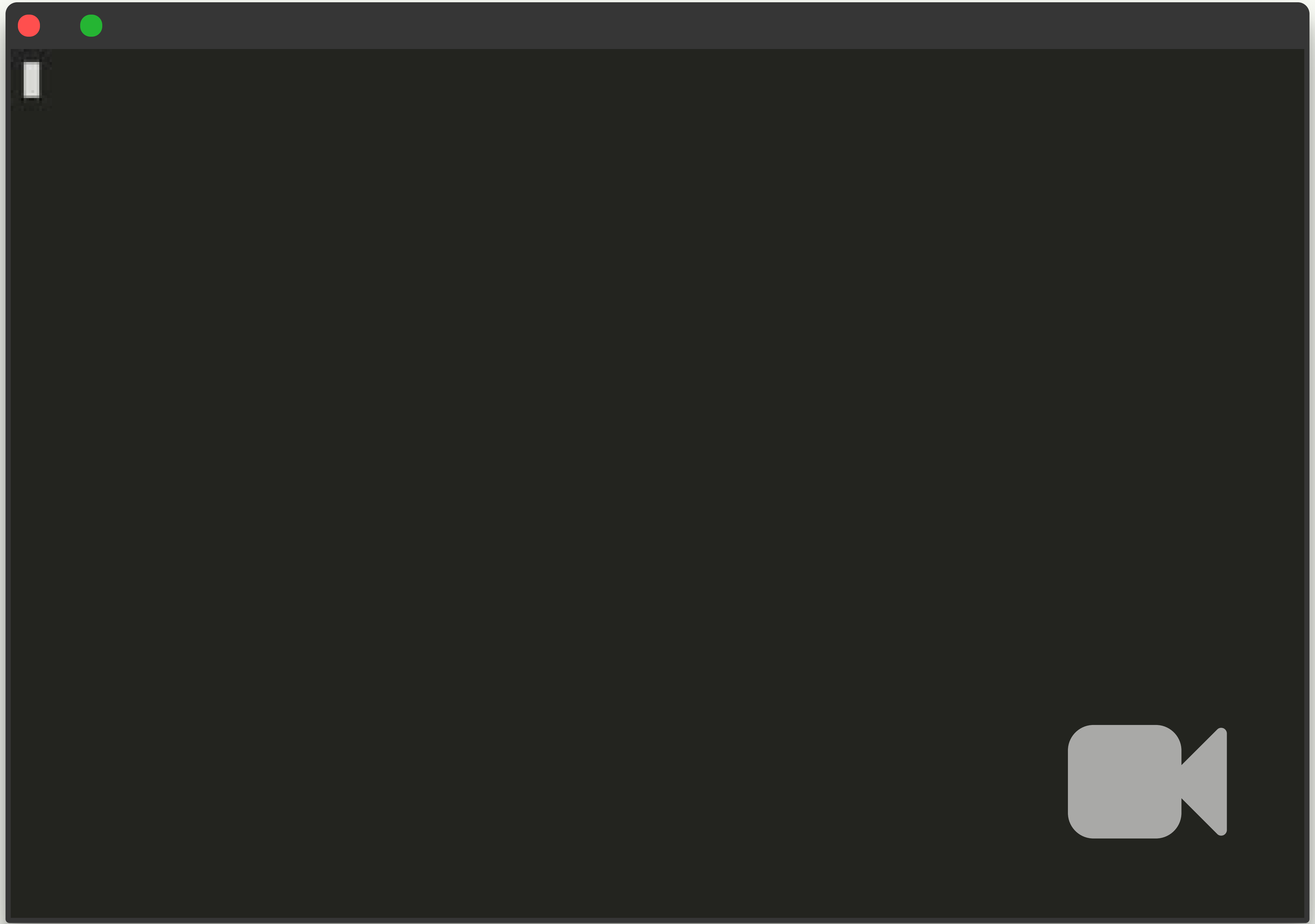
```
<--- Last few GCs --->
[10915:0x3f6b950]    2244 ms: Mark-sweep 30.8 (70.4) -> 30.8 (70.4) MB, 19.2 / 0.0 ms
 allocation failure GC in old space requested
[10915:0x3f6b950]    2269 ms: Mark-sweep 30.8 (70.4) -> 30.8 (39.4) MB, 25.5 / 0.0 ms
 last resort
[10915:0x3f6b950]    2294 ms: Mark-sweep 30.8 (39.4) -> 30.8 (39.4) MB, 24.3 / 0.0 ms
 last resort

FATAL ERROR: CALL_AND_RETRY_LAST Allocation failed - JavaScript heap out of memory

Aborted (core dumped)
```

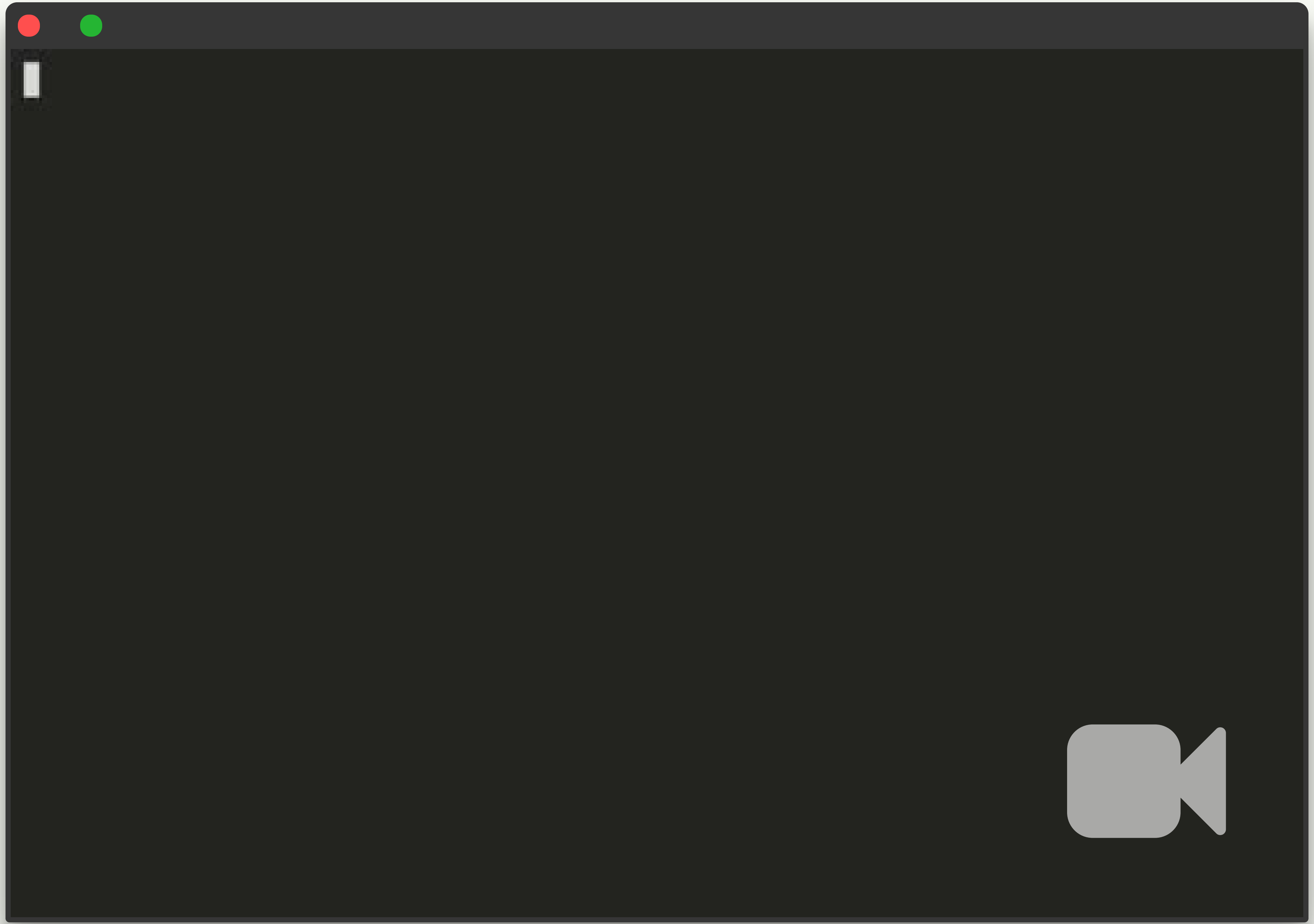
Press any key to continue _

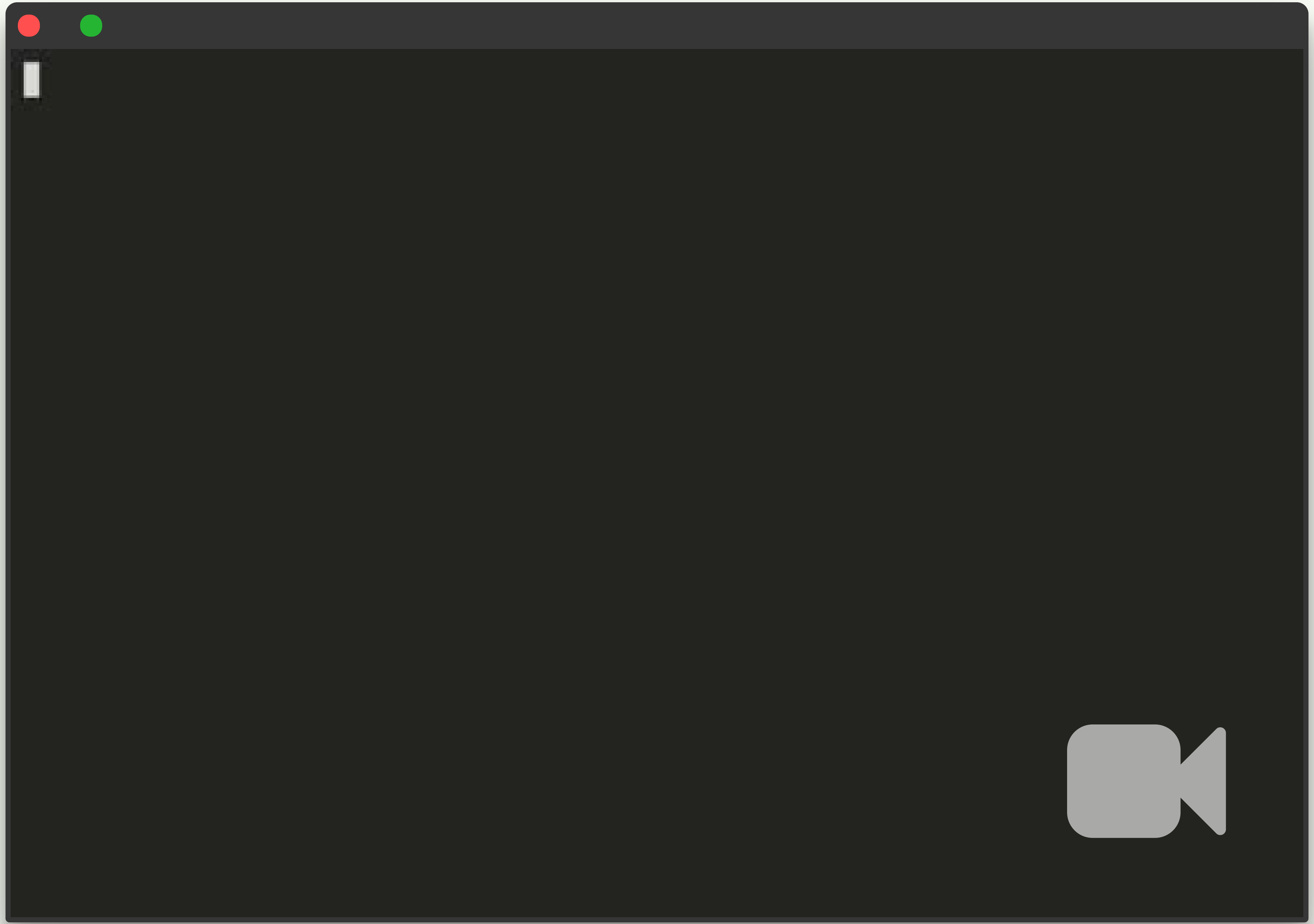


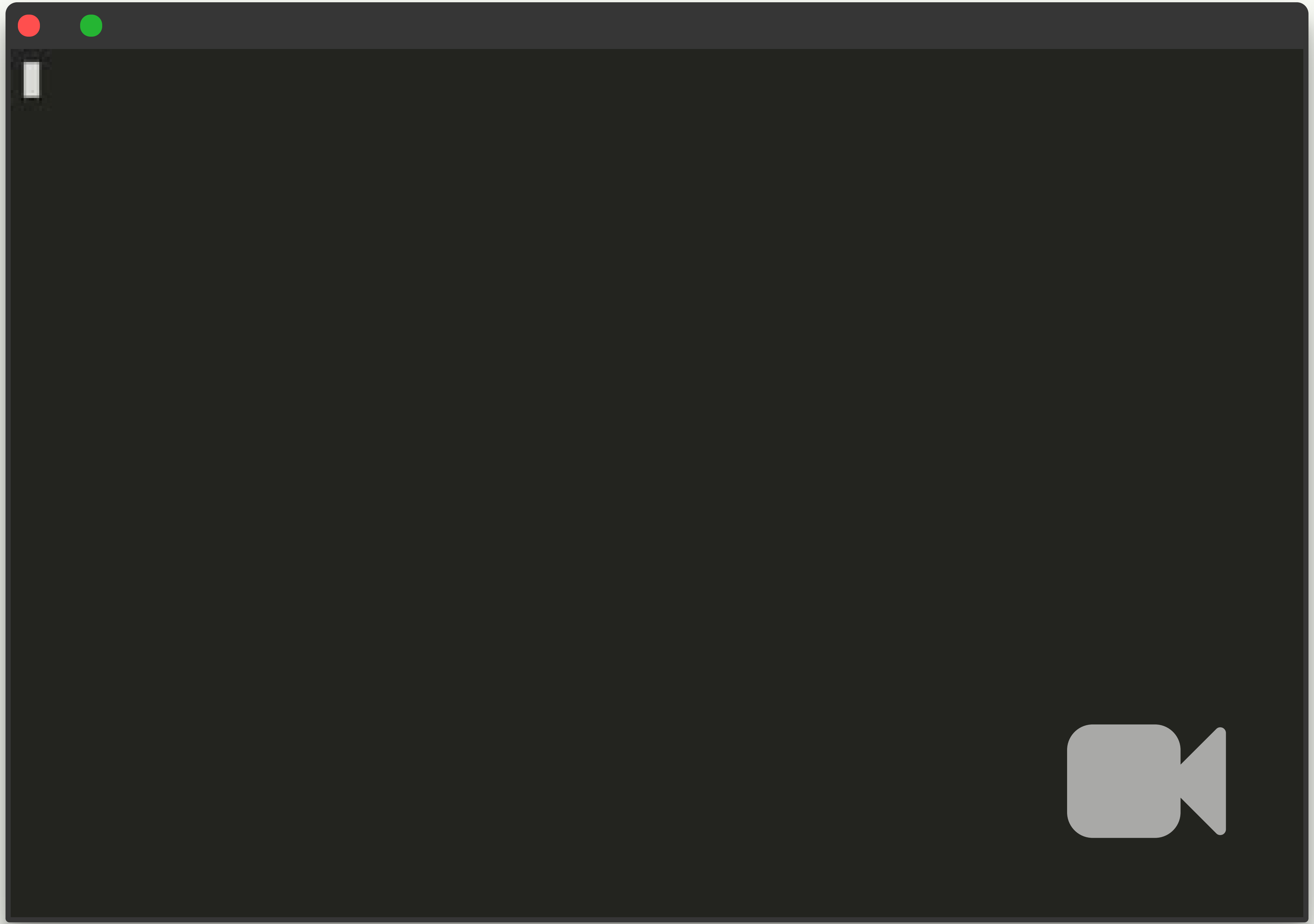


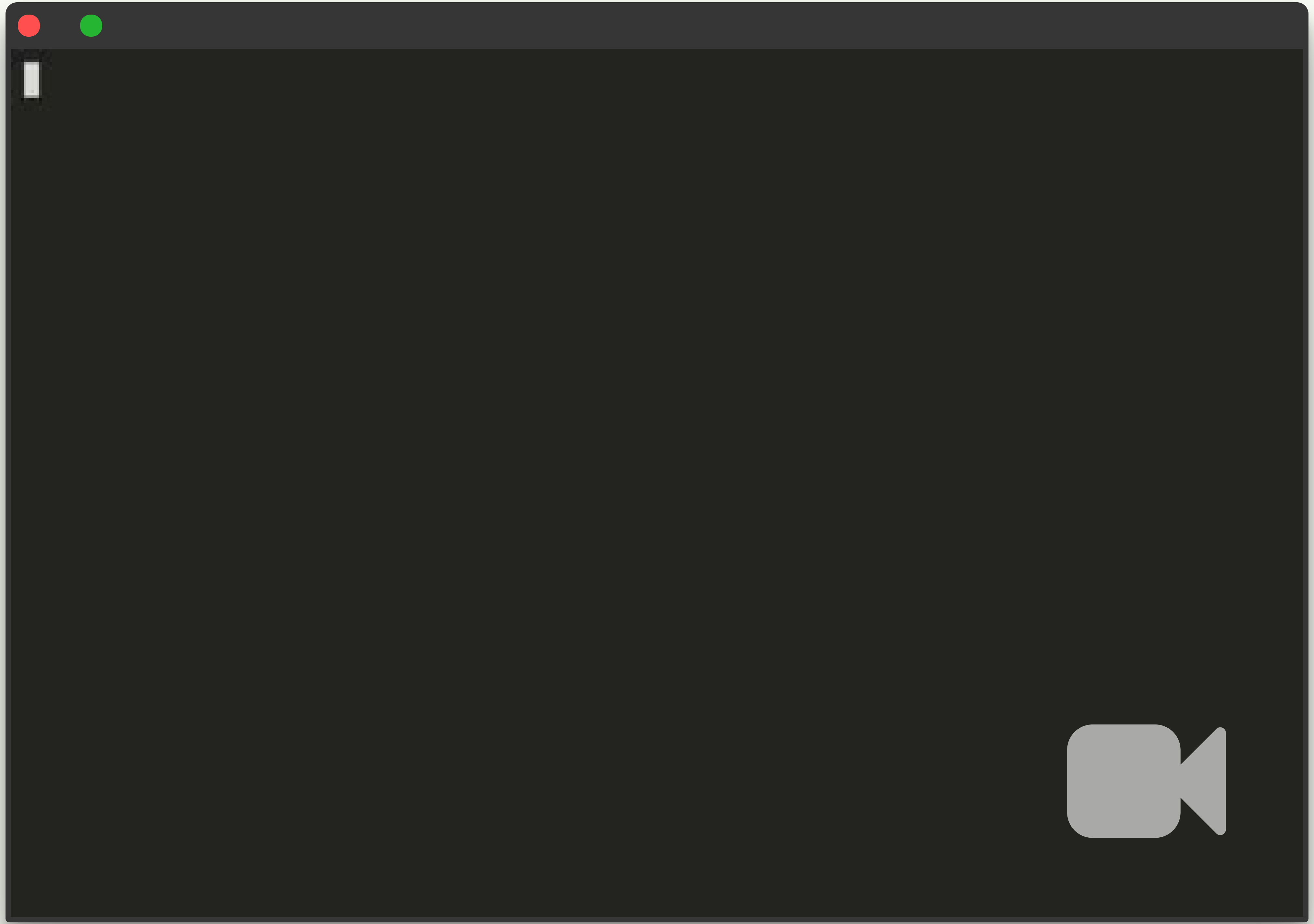


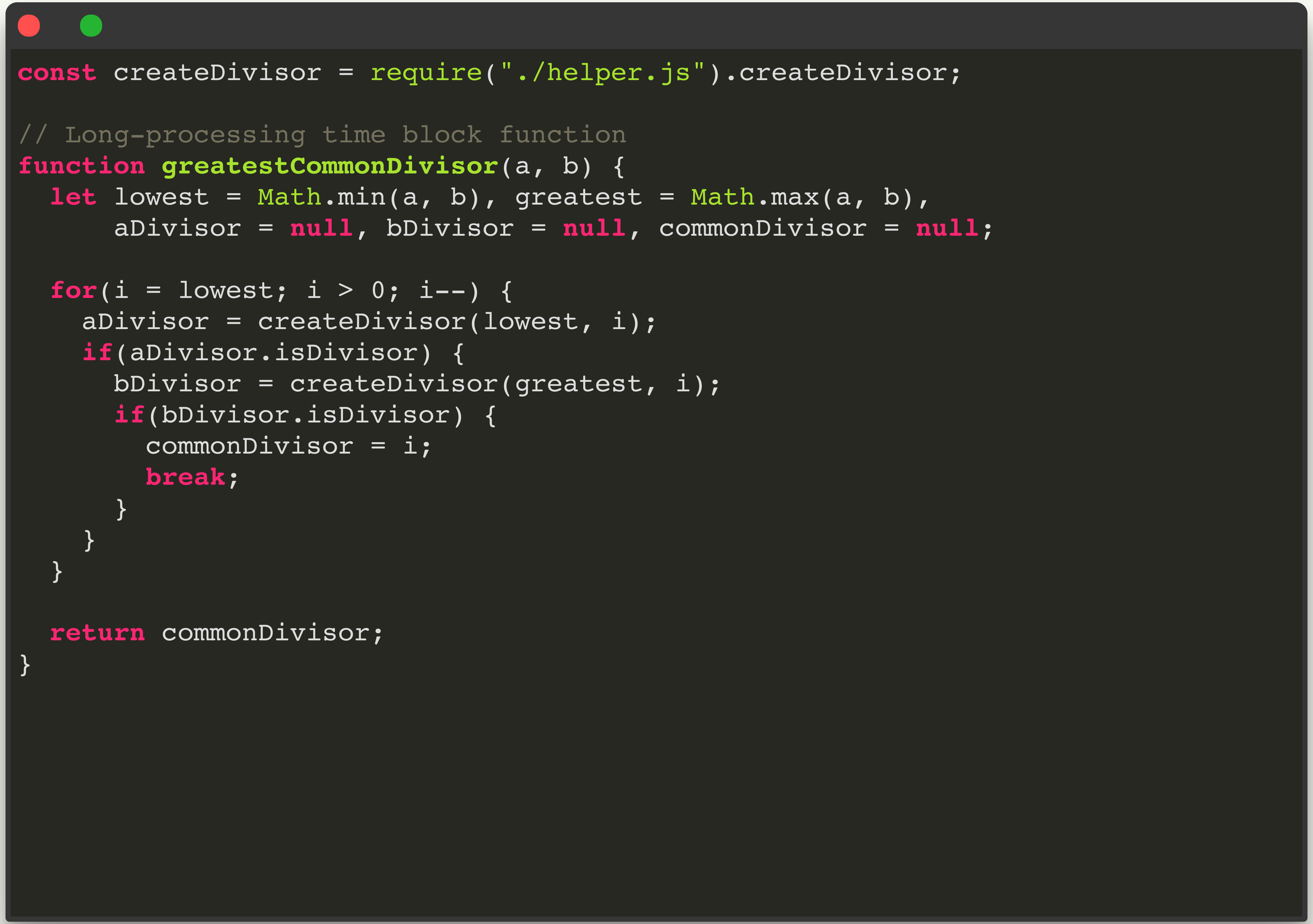






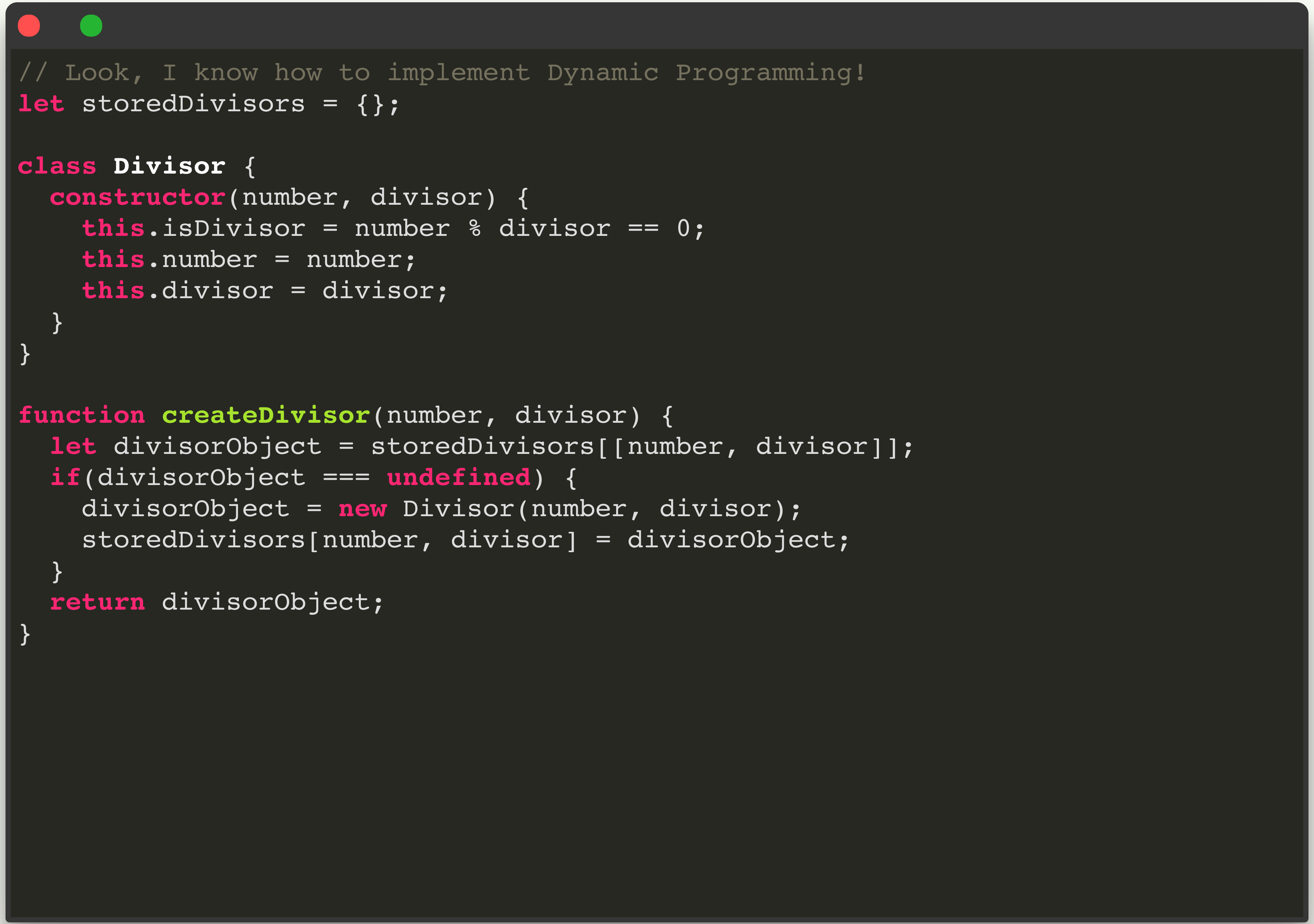






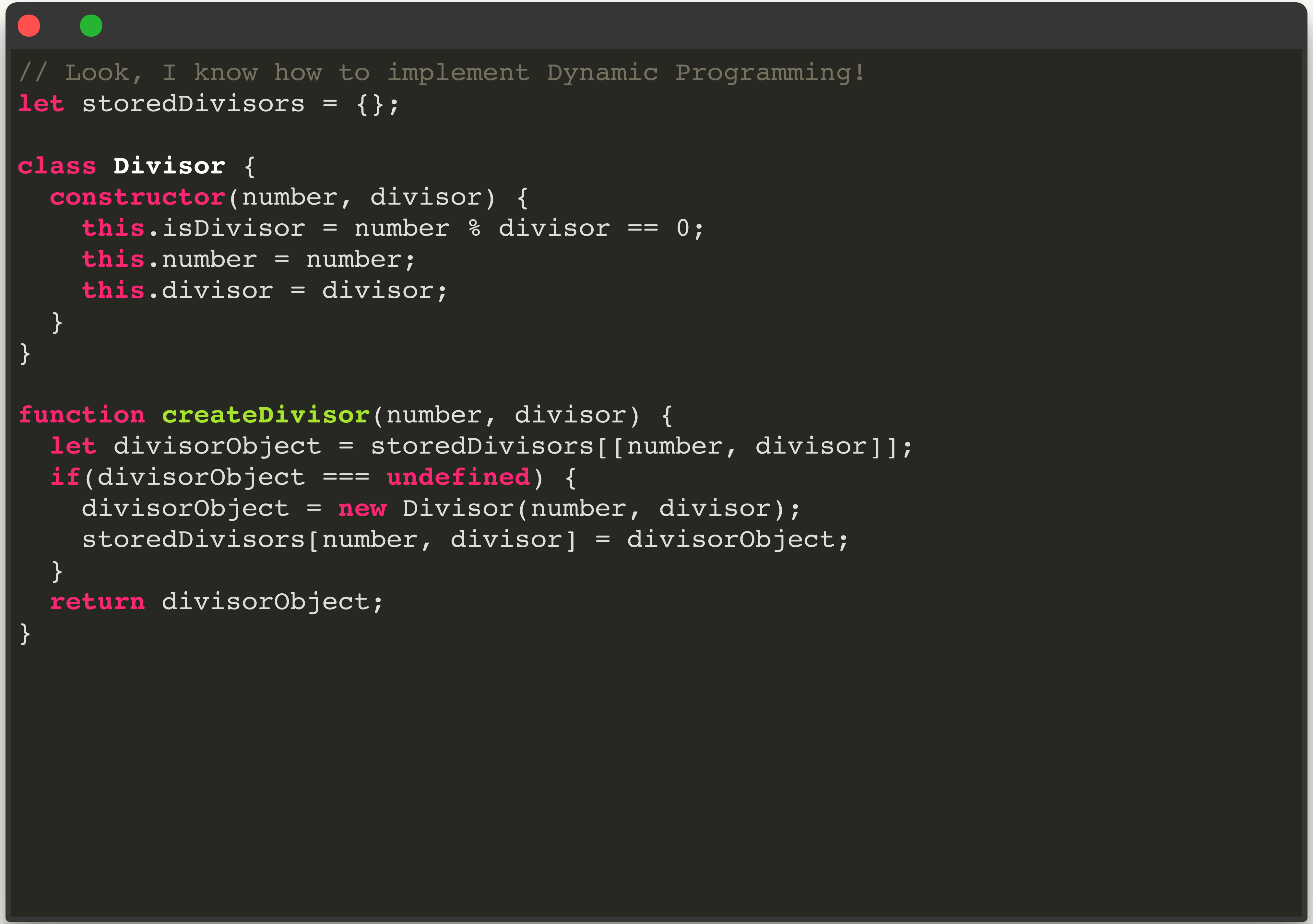
```
const createDivisor = require("./helper.js").createDivisor;

// Long-processing time block function
function greatestCommonDivisor(a, b) {
  let lowest = Math.min(a, b), greatest = Math.max(a, b),
      aDivisor = null, bDivisor = null, commonDivisor = null;

  for(i = lowest; i > 0; i--) {
    aDivisor = createDivisor(lowest, i);
    if(aDivisor.isDivisor) {
      bDivisor = createDivisor(greatest, i);
      if(bDivisor.isDivisor) {
        commonDivisor = i;
        break;
      }
    }
  }

  return commonDivisor;
}
```

problematic/index.js



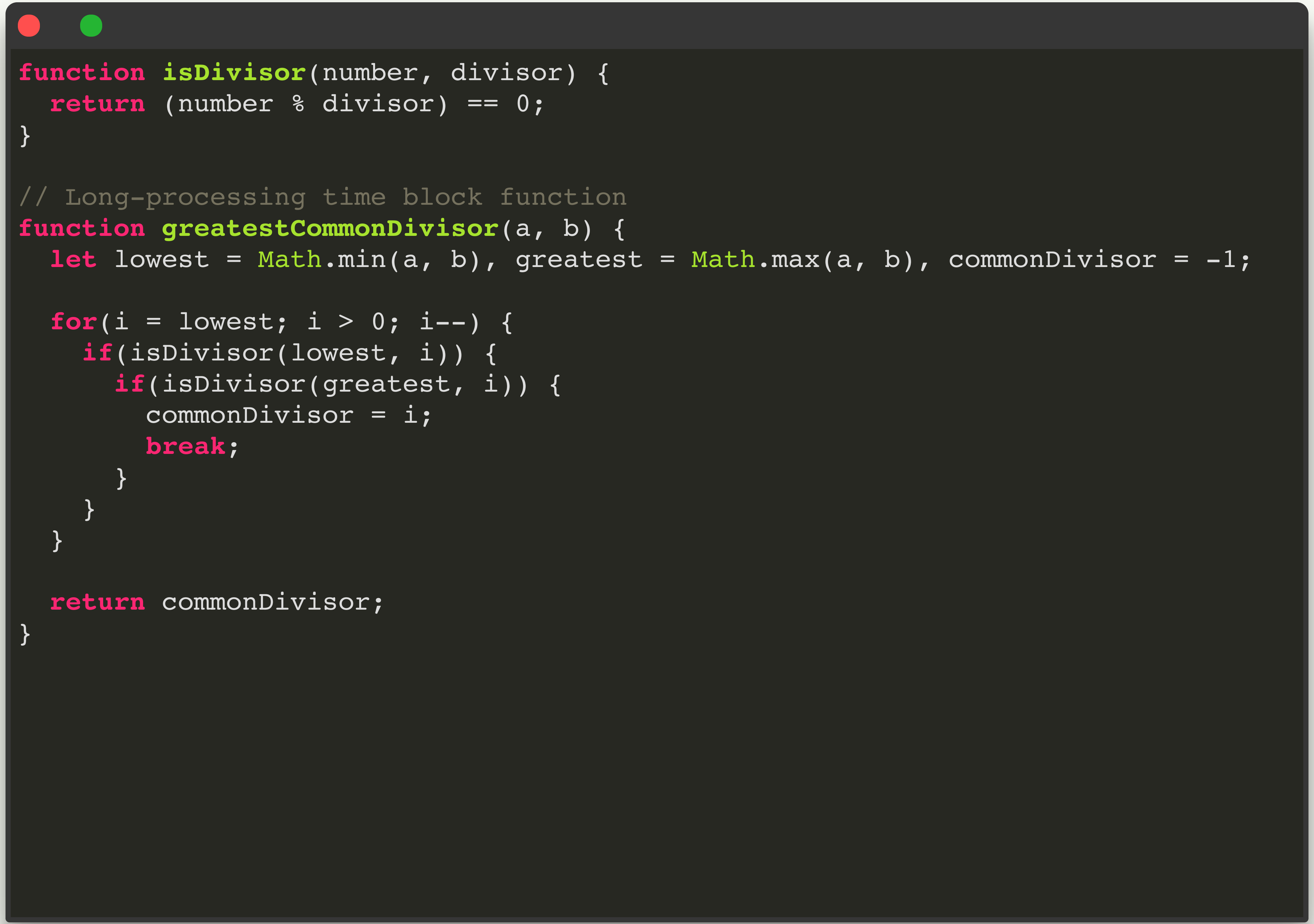
```
// Look, I know how to implement Dynamic Programming!
let storedDivisors = {};

class Divisor {
  constructor(number, divisor) {
    this.isDivisor = number % divisor == 0;
    this.number = number;
    this.divisor = divisor;
  }
}

function createDivisor(number, divisor) {
  let divisorObject = storedDivisors[[number, divisor]];
  if(divisorObject === undefined) {
    divisorObject = new Divisor(number, divisor);
    storedDivisors[number, divisor] = divisorObject;
  }
  return divisorObject;
}
```

problematic/helper.js



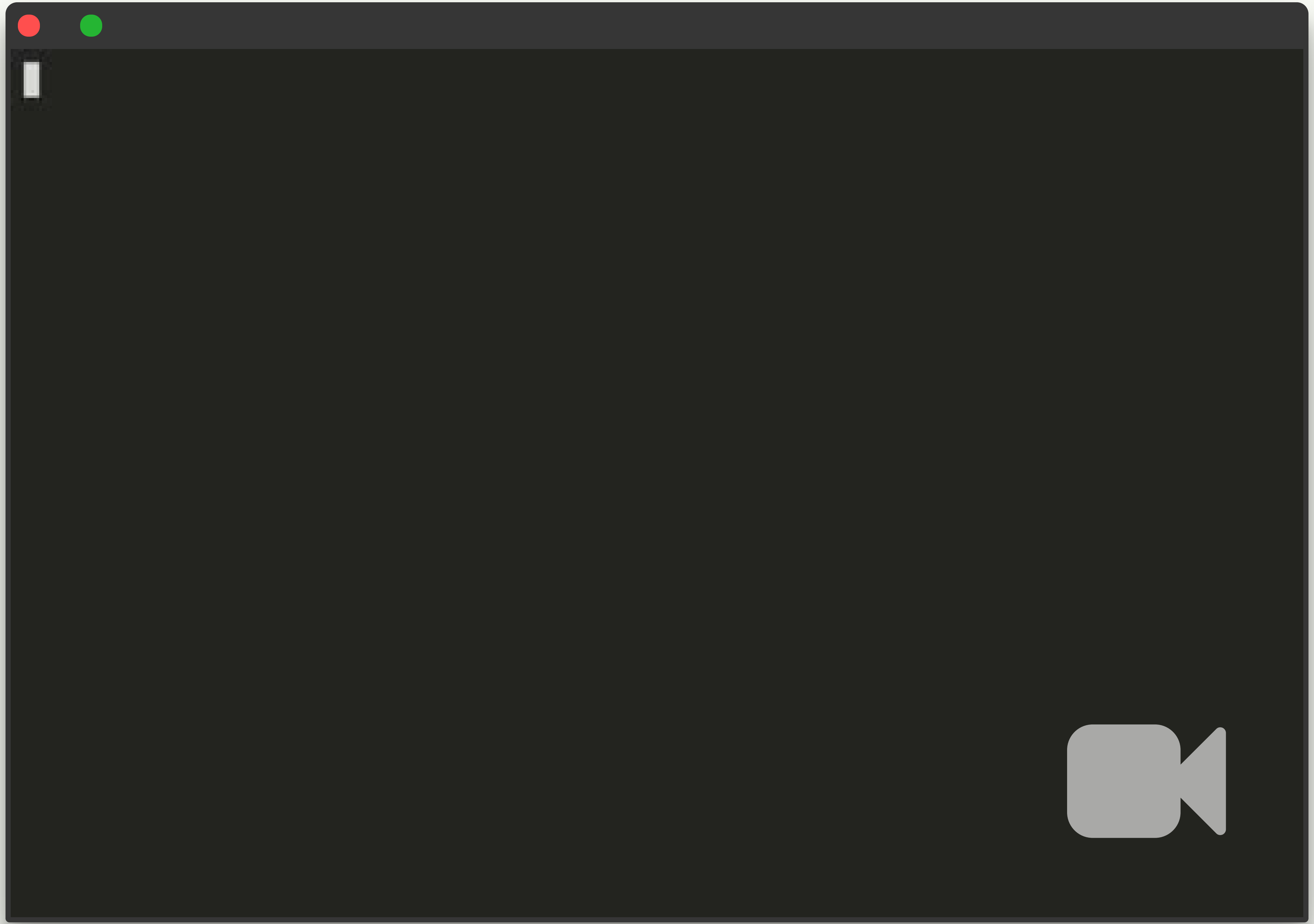
```javascript
// Look, I know how to implement Dynamic Programming!
let storedDivisors = {};

class Divisor {
  constructor(number, divisor) {
    this.isDivisor = number % divisor == 0;
    this.number = number;
    this.divisor = divisor;
  }
}

function createDivisor(number, divisor) {
  let divisorObject = storedDivisors[[number, divisor]];
  if(divisorObject === undefined) {
    divisorObject = new Divisor(number, divisor);
    storedDivisors[number, divisor] = divisorObject;
  }
  return divisorObject;
}
```

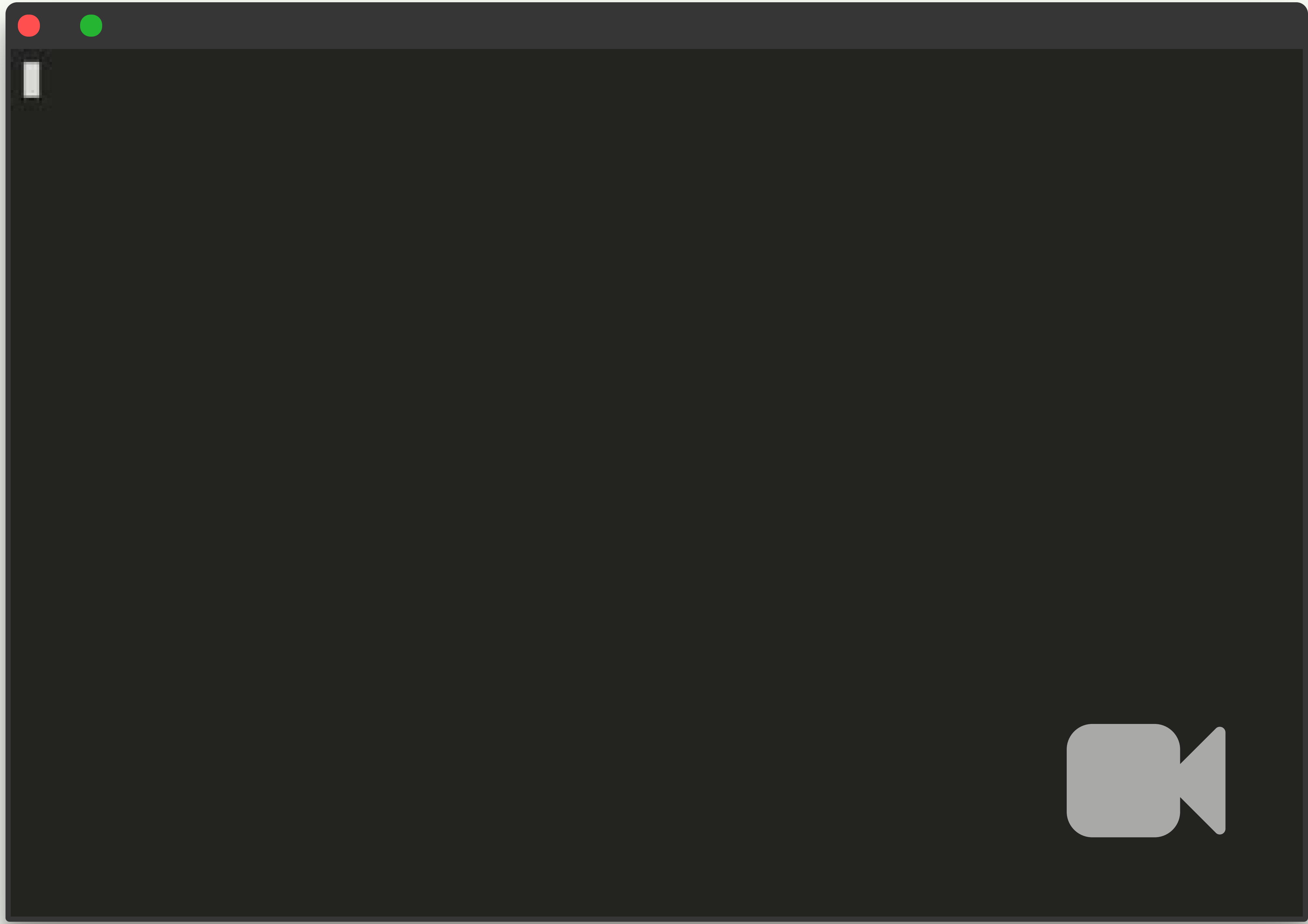
problematic/helper.js



```
function isDivisor(number, divisor) {
  return (number % divisor) == 0;
}

// Long-processing time block function
function greatestCommonDivisor(a, b) {
  let lowest = Math.min(a, b), greatest = Math.max(a, b), commonDivisor = -1;

  for(i = lowest; i > 0; i--) {
    if(isDivisor(lowest, i)) {
      if(isDivisor(greatest, i)) {
        commonDivisor = i;
        break;
      }
    }
  }

  return commonDivisor;
}
```

problematic/index.js





sthima

https://www.sthima.com/

@mmarkini

http://mmarchini.me/

oss@mmarchini.me

https://github.com/mmarchini/

sthima

https://www.sthima.com/

We're Hiring!

@mmarkini

http://mmarchini.me/

oss@mmarchini.me

https://github.com/mmarchini/