

# DEBUGGING

PREVENTION  
DETECTION  
DIAGNOSIS  
FIXING



We're back. We've covered prevention, detection, and installed some great tools.

# DETECTION

- 🐛 Program crashes or “bugs out”
- 🐛 Build/compile error
- 🐛 Automated testing fails

How do you know when there is a bug?

Sometimes it's obvious. But you'll need to keep your eyes open and your ears to the ground.

Testing: It is difficult to overstate the tremendous value of automated testing in general.

ALWAYS read the output of a program on the command line

# DIAGNOSIS

Time to diagnose.



## **MANDATORY**



Browser console open



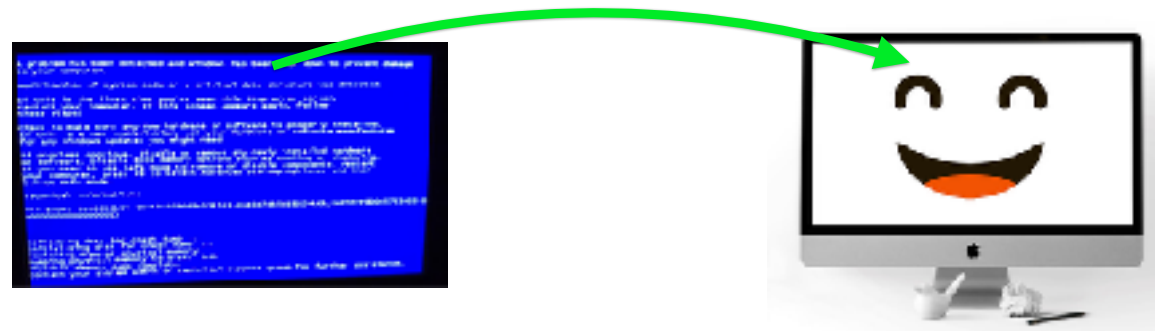
Server console open



Read the stack trace

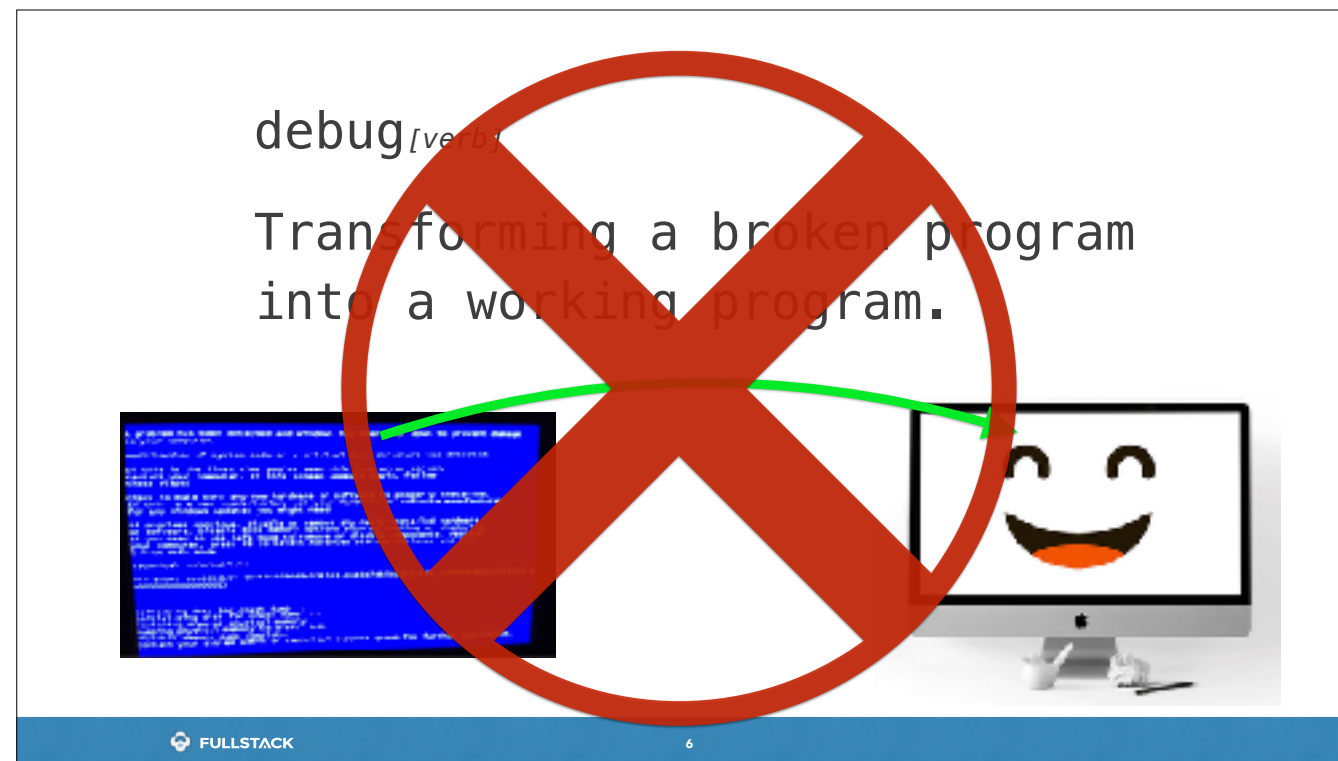
debug<sub>[verb]</sub>

Transforming a broken program  
into a working program.



This is one way to think about debugging.

But it isn't the best one.



“A change in perspective is worth eighty IQ points.”  
— Alan Kay ( He invented the personal computer, have you heard of him? )

You understand the program will  
run in a particular way.

***But you are wrong.***

**Bug!**



Debugging isn't about fixing the program.

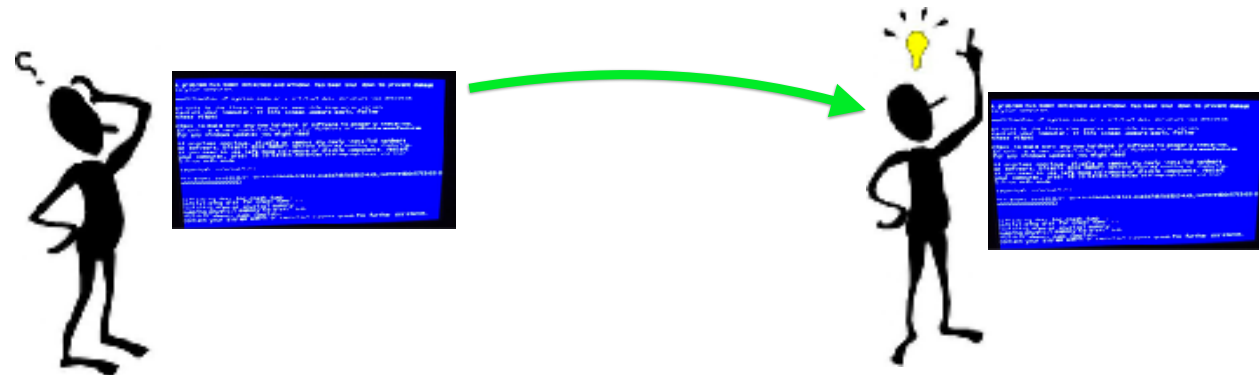
Debugging is about fixing your understanding.

Debugging is about finding and disabusing yourself of false assumptions.

Once your understanding is rectified, fixing the program will  
occur simply.

debug<sub>[verb]</sub>

Transforming a **broken understanding**  
of program into a **working understanding**.



This is a better way to think of debugging.

Debugging is first targeted towards yourself and your understanding of the program, not the program itself.



# Verbalizing Helps



# Sleeping Helps



Write amazing console.logs

```
console.log(  
  "very useful context",  
  reallyImportantObject  
)
```

The answer isn't always exactly where the output says it is, but it usually aims you in the general direction.

“unexpected token” -> misspelled `import`

## Learn to read stack traces

```
✖ ▶ Uncaught Error: Read the stacktrace!  
  at slide (index.js:3)  
  at combineReducers.js:40  
  at Array.forEach (<anonymous>)  
  at assertReducerShape (combineReducers.js:38)  
  at combineReducers (combineReducers.js:94)  
  at configureStore (configure-store.js:15)  
  at Object.getStore (bootloader.js:27)  
  at Object.renderApp (bootloader.js:34)
```

## Learn to read stack traces

```
ERROR in ./src/reducers/slide/index.js
Module build failed: SyntaxError: Unexpected token, expected ; (1:7)

> 1 | import Types from './types'
    |         ^
  2 | export default function slide (prevState = null, action) {
  3 |   switch (action.type) {
  4 |     case Types.EDIT_SLIDE:

@ ./src/reducers/index.js 8:14-32
@ ../anecdote/src/bootloader.js
@ ./src/index.js
```

The answer isn't always exactly where the output says it is, but it usually aims you in the general direction.

“unexpected token” -> misspelled `import`

## Learn to read stack traces

```
ERROR in ./src/reducers/slide/index.js
Module build failed: SyntaxError: Unexpected token, expected ; (1:7)

> 1 | import Types from './types'
    |         ^
  2 | export default function slide (prevState = null, action) {
  3 |   switch (action.type) {
  4 |     case Types.EDIT_SLIDE:

@ ./src/reducers/index.js 8:14-32
@ ../anecdote/src/bootloader.js
@ ./src/index.js
```

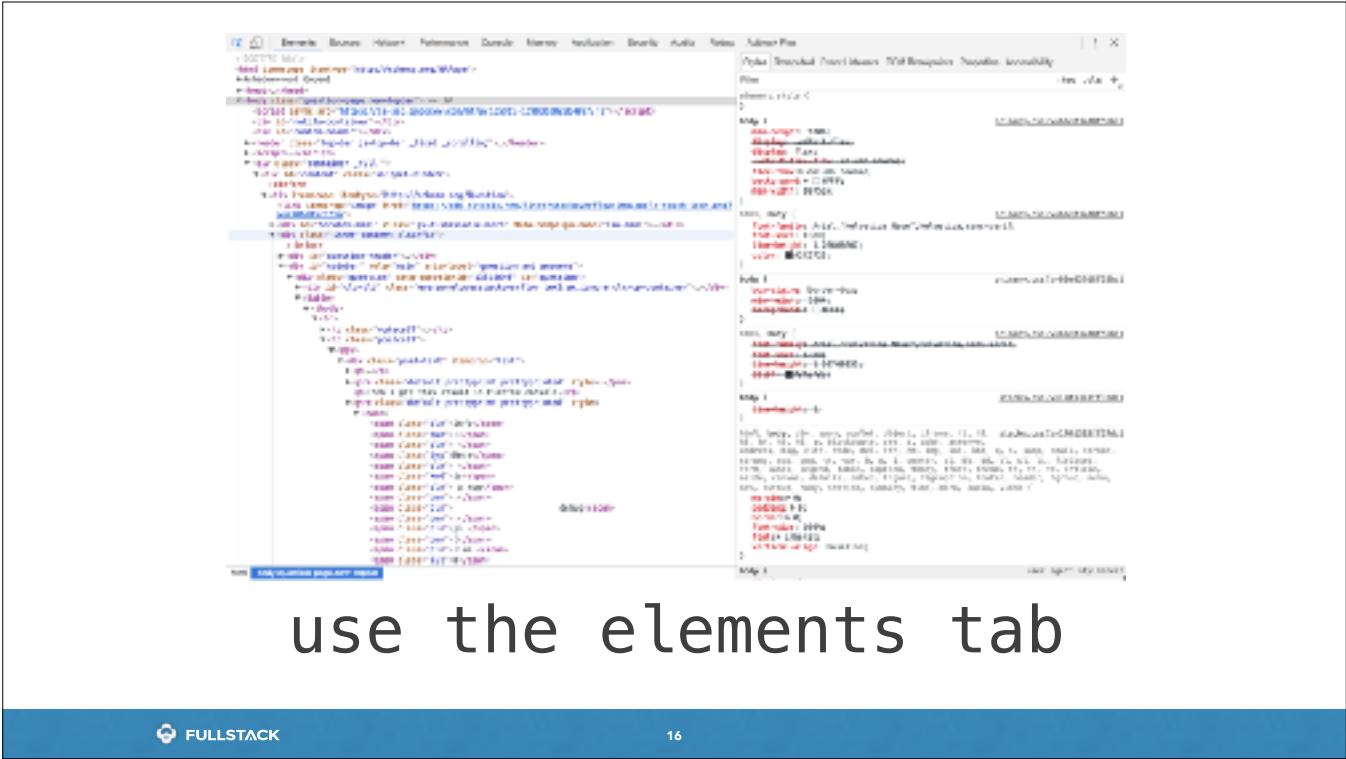
The answer isn't always exactly where the output says it is, but it usually aims you in the general direction.

“unexpected token” -> misspelled `import`



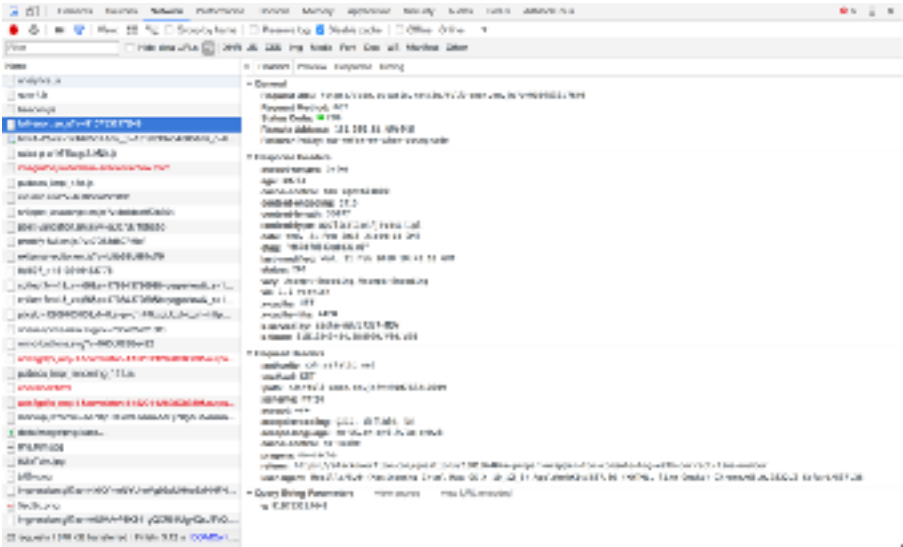
use the debugger

Make a habit of keeping your console open.



Make a habit of keeping your console open.

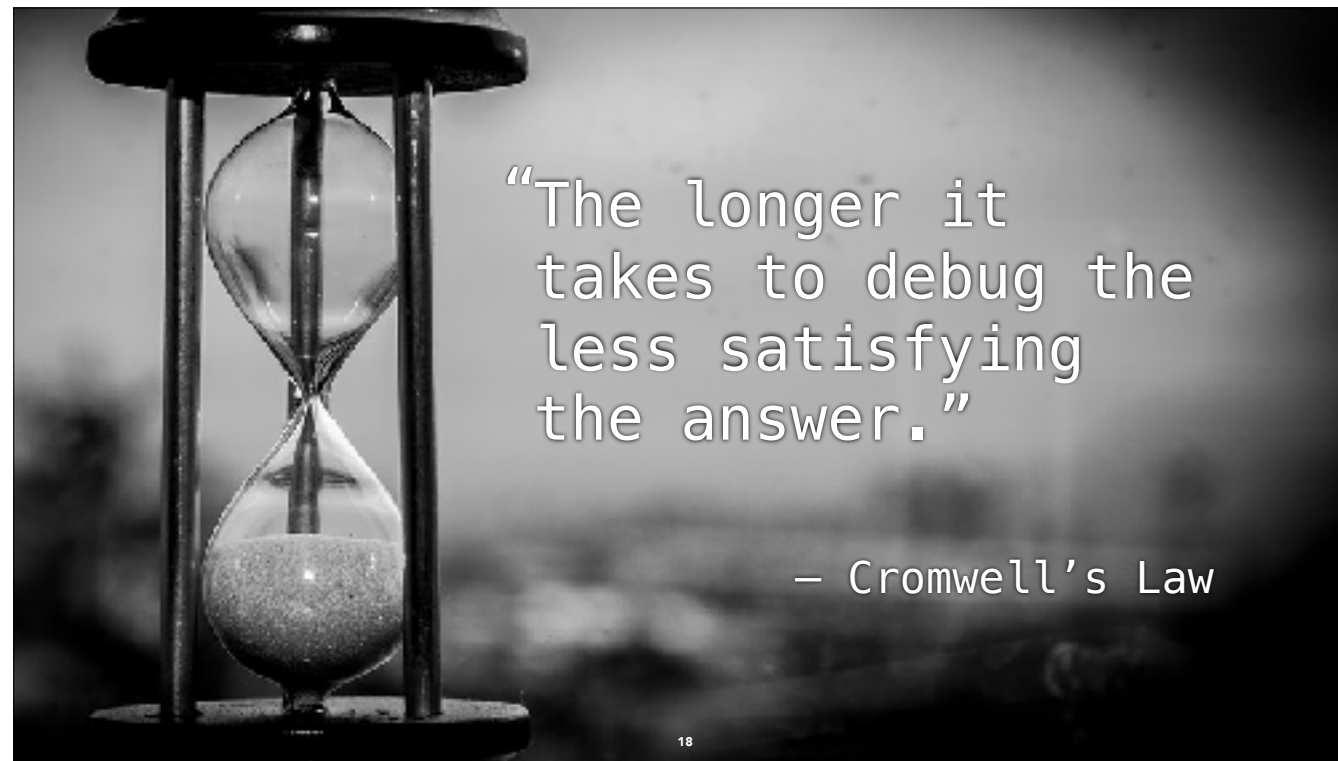




The screenshot shows the Redux DevTools application with the 'Network' tab selected. The left pane displays a list of network requests, with the first one, 'GET /api/users', highlighted. The right pane shows the details of this request, including the status (200), headers, and the JSON response body. The response body is a list of user objects, each containing a name, email, and password. The bottom of the image features a blue banner with the text 'use the network tab' and the Fullstack logo.

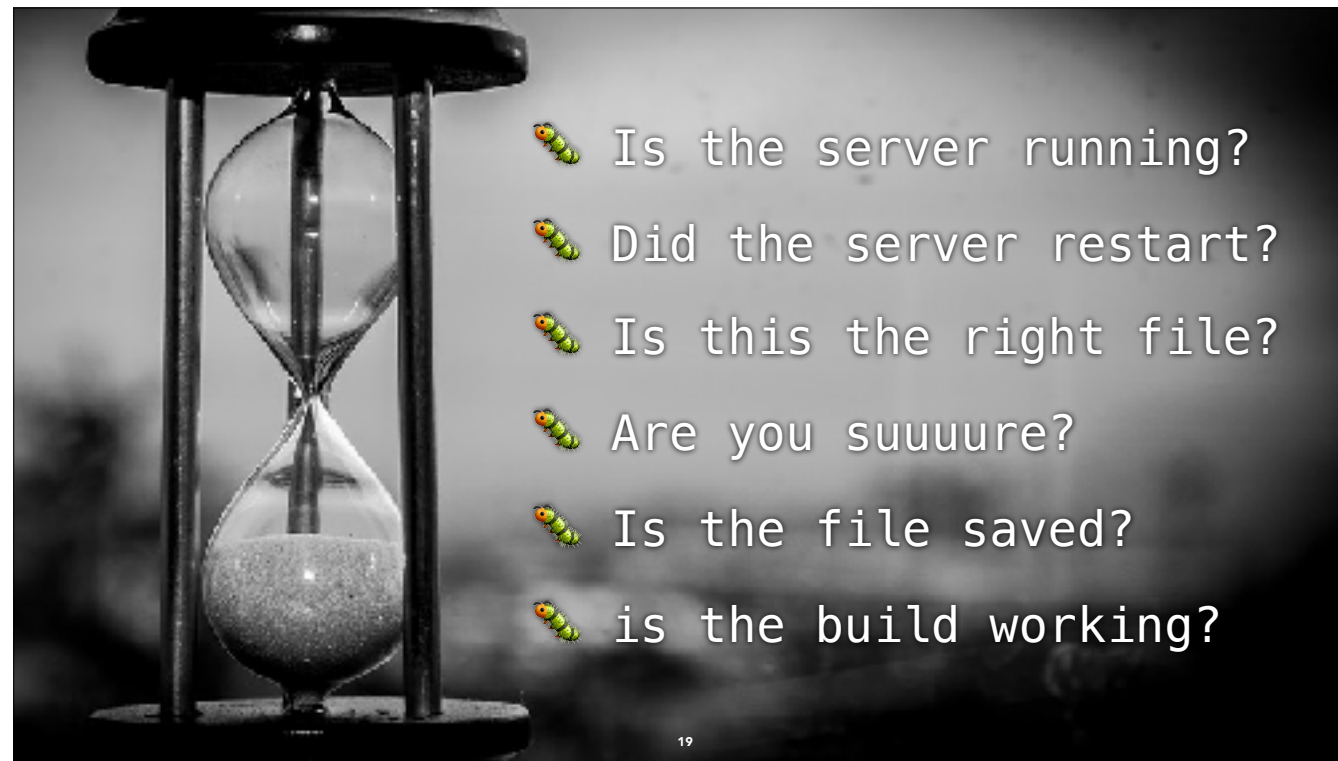
use the network tab

Make a habit of keeping your console open.



Sometimes debugging feels like nothing you do could be right.

Try a sanity check.



Sometimes debugging feels like nothing you do could be right.

Try a sanity check.

What You've Learned Today

BUG

- Prevention
- Detection
- Diagnosis
- Fixing

FULLSTACK 20

Now, you'll see that we didn't really cover "fixing". Once you've diagnosed the issue, fixing is the easy part. Understanding the bug means you have the knowledge to address it.

This is why...

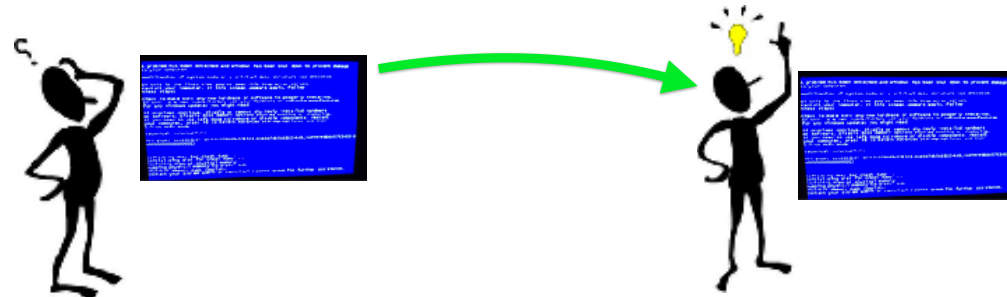


If you only remember one thing



`debug`<sub>[verb]</sub>

Transforming a **broken understanding**  
of program into a **working understanding**.



If you remember only one thing, it's this.

The hard work of debugging is going on inside your head.



If you remember three things



**MANDATORY**



Browser console open



Server console open



Read the stack trace