

Programming Fundamentals

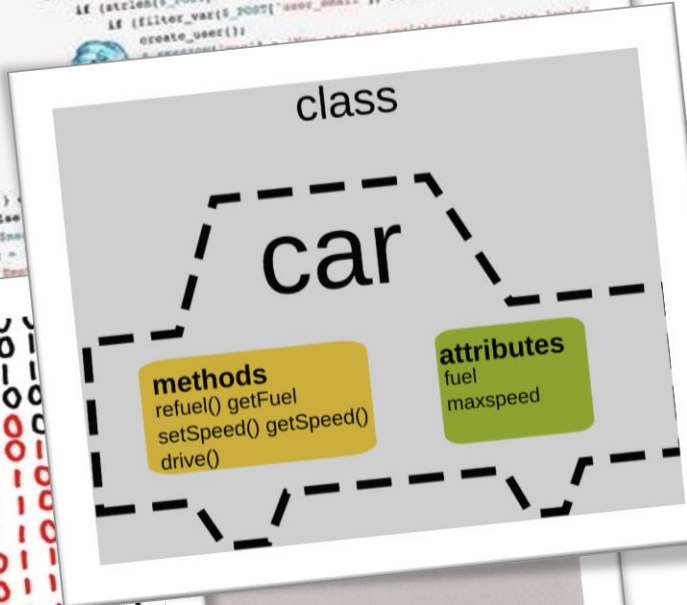
Stefano Balietti

Center for European Social Science Research at Mannheim University (MZES)
Alfred-Weber Institute of Economics at Heidelberg University

@balietti | stefanobalietti.com | @nodegameorg | nodegame.org



Building Digital Skills: 12-13 March 2020, University of Luzern



Hands On: Node.JS Console



Open the Node.JS console:

Windows: Git Bash

Mac / Linux: Terminal

```
balistef@mzes072 MINGW64 ~  
$ node  
Welcome to Node.js v12.15.0.  
Type ".help" for more information.  
> >
```

Hands On 2: Node.JS Console



```
> function foo() {  
... console.log(a);  
... }
```

```
undefined
```

```
> let a = 'bar';
```

```
undefined
```

```
> foo();
```

```
'bar'
```

```
undefined
```

```
>
```

- Multiline function declaration (...)
- Variable a can be declared and assigned anytime before function it is executed
- Undefined means that there is no return value

Hands On 2: Node.JS Console



```
> function foo() {  
... console.log(a);  
... }
```

```
undefined
```

```
> let a = 'bar';
```

```
undefined
```

```
> foo();
```

```
'bar'
```

```
undefined
```

```
>
```

- What is this **foo** and **bar** thingy?

<https://en.wikipedia.org/wiki/Foobar>

Hands On 2: Node.JS Console



```
function foo() {  
    console.log(a);  
}  
let a = 'bar';  
foo();  
let a = 'BAR';  
foo();
```

- Update the value of a printed by foo

Hands On 2: Node.JS Console



```
function foo() {  
  console.log(a);  
}
```

```
let a = 'bar';  
foo();
```

```
let a = 'BAR';  
foo();
```

- ES6 let operator does not permit to redeclare a variable with the same name

Thrown:
SyntaxError: Identifier 'a'
has already been declared

Hands On 2: Node.JS Console



```
function foo() {  
    console.log(a);  
}  
let a = 'bar';  
foo();  
a = 'BAR';  
foo();
```

- Ctrl-C twice to exit the Node.JS console

Hands On 3: Launch a Node Program



Create file `hello.js` with your text editor.

```
const fs = require('fs');
const path = require('path');

let myfile = path.join(__dirname, 'out.txt');
fs.writeFileSync(myfile, 'Hello Pilsen');
```

Save it and then execute it in the terminal

```
balistef@mzes072 MINGW64 ~/www/nodegame-workshop (master)
$ node hello.js
```


Understanding Errors



Create file constant-error.js

```
const fs = require('fs');  
const path = require('path');
```

```
// Assign a new property to the fs object.  
fs.aNewProperty = 'some value';  
// Reassign the fs object.  
fs = 'a new life';
```

 Only one of the two assignments is valid. Which one?

Understanding Errors



```
balistef@mzes072 MINGW64 ~/www/nodegame-workshop (master)
$ node constant-error.js
C:\Users\balistef\www\nodegame-workshop\constant-error.js:7
fs = 'a new value';
  ^
TypeError: Assignment to constant variable.
    at Object.<anonymous> (C:\Users\balistef\www\nodegame-workshop\constant-error.js:7:4)
    at Module._compile (internal/modules/cjs/loader.js:776:30)
    at Object.Module._extensions..js (internal/modules/cjs/loader.js:787:10)
    at Module.load (internal/modules/cjs/loader.js:653:32)
    at tryModuleLoad (internal/modules/cjs/loader.js:593:12)
    at Function.Module._load (internal/modules/cjs/loader.js:585:3)
    at Function.Module.runMain (internal/modules/cjs/loader.js:829:12)
    at startup (internal/bootstrap/node.js:283:19)
    at bootstrapNodeJSCore (internal/bootstrap/node.js:622:3)
```

Understanding Errors



```
balistef@mzes072 MINGW64 ~/www/nodegame-workshop (master)
$ node constant-error.js
C:\Users\balistef\www\nodegame-workshop\constant-error.js:7
fs = 'a new value';
  ^
```

Code causing error and error message

```
TypeError: Assignment to constant variable.
```

```
at Object.<anonymous> (C:\Users\balistef\www\nodegame-workshop\constant-error.js:7:4)
at Module._compile (internal/modules/cjs/loader.js:776:30)
at Object.Module._extensions..js (internal/modules/cjs/loader.js:717:10)
at Module.load (internal/modules/cjs/loader.js:653:32)
at tryModuleLoad (internal/modules/cjs/loader.js:593:12)
at Function.Module._load (internal/modules/cjs/loader.js:529:12)
at Function.Module.runMain (internal/modules/cjs/loader.js:829:12)
at startup (internal/bootstrap/node.js:283:19)
at bootstrapNodeJSCore (internal/bootstrap/node.js:622:3)
```

Constants are not immutable, are just not reassignable.

Understanding Errors



```
balistef@mzes072 MINGW64 ~/www/nodegame-workshop (master)
$ node constant-error.js
C:\Users\balistef\www\nodegame-workshop\constant-error.js:7
fs = 'a new value';
  ^
```

File name causing the error

```
TypeError: Assignment to constant variable.
    at Object.<anonymous> (C:\Users\balistef\www\nodegame-workshop\constant-error.js:7:4)
    at Module._compile (internal/modules/cjs/loader.js:776:30)
    at Object.Module._extensions..js (internal/modules/cjs/loader.js:787:10)
    at Module.load (internal/modules/cjs/loader.js:653:32)
    at tryModuleLoad (internal/modules/cjs/loader.js:593:12)
    at Function.Module._load (internal/modules/cjs/loader.js:585:3)
    at Function.Module.runMain (internal/modules/cjs/loader.js:829:12)
    at startup (internal/bootstrap/node.js:283:19)
    at bootstrapNodeJSCore (internal/bootstrap/node.js:622:3)
```

Understanding Errors



```
balistef@mzes072 MINGW64 ~/www/nodegame-workshop (master)
$ node constant-error.js
C:\Users\balistef\www\nodegame-workshop\constant-error.js:7
fs = 'a new value';
  ^
```

Line Number and Column Number

```
TypeError: Assignment to constant variable.
    at Object.<anonymous> (C:\Users\balistef\www\nodegame-workshop\constant-error.js:7:4)
    at Module._compile (internal/modules/cjs/loader.js:776:30)
    at Object.Module._extensions..js (internal/modules/cjs/loader.js:787:10)
    at Module.load (internal/modules/cjs/loader.js:653:32)
    at tryModuleLoad (internal/modules/cjs/loader.js:593:12)
    at Function.Module._load (internal/modules/cjs/loader.js:585:3)
    at Function.Module.runMain (internal/modules/cjs/loader.js:829:12)
    at startup (internal/bootstrap/node.js:283:19)
    at bootstrapNodeJSCore (internal/bootstrap/node.js:622:3)
```

Understanding Errors



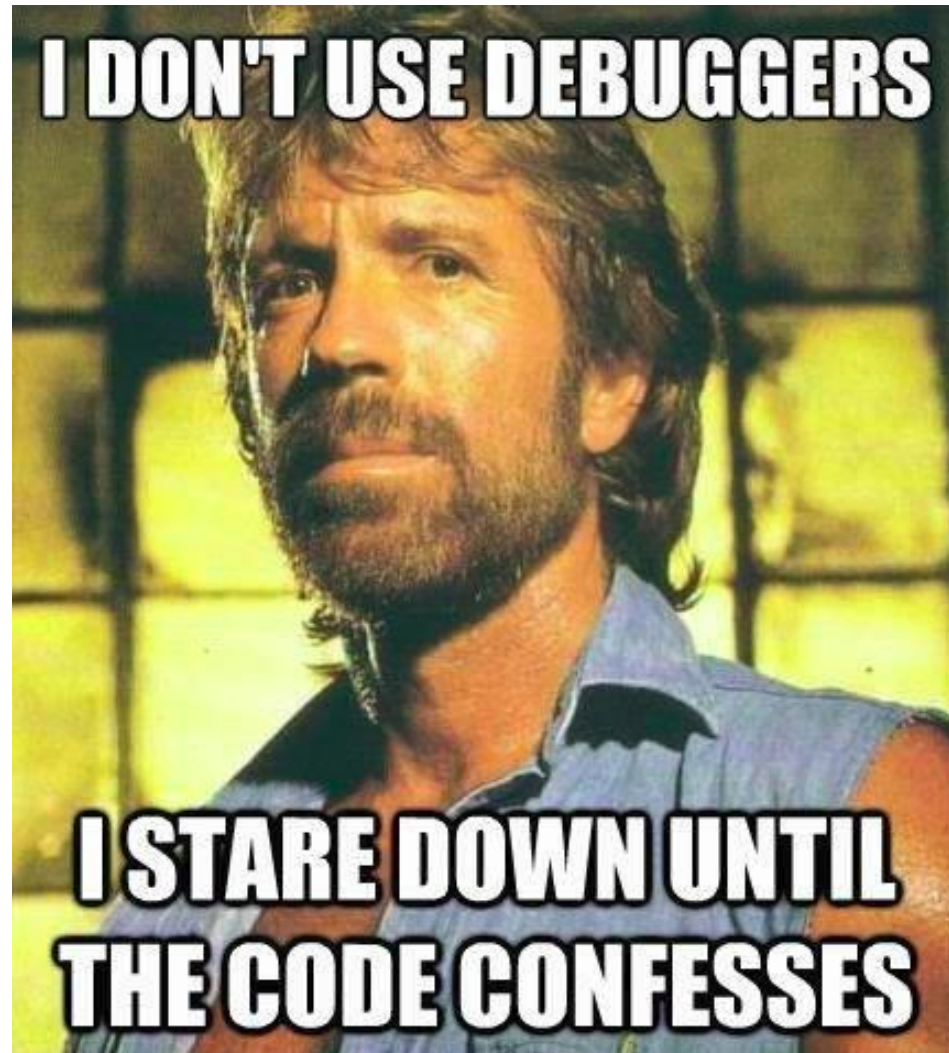
```
balistef@mzes072 MINGW64 ~/www/nodegame-workshop (master)
$ node constant-error.js
C:\Users\balistef\www\nodegame-workshop\constant-error.js:7
fs = 'a new value';
  ^
```

Stack Trace

TypeError: Assignment to constant variable.

```
at Object.<anonymous> (C:\Users\balistef\www\nodegame-workshop\constant-error.js:7:4)
at Module._compile (internal/modules/cjs/loader.js:776:30)
at Object.Module._extensions..js (internal/modules/cjs/loader.js:787:10)
at Module.load (internal/modules/cjs/loader.js:653:32)
at tryModuleLoad (internal/modules/cjs/loader.js:593:12)
at Function.Module._load (internal/modules/cjs/loader.js:585:3)
at Function.Module.runMain (internal/modules/cjs/loader.js:829:12)
at startup (internal/bootstrap/node.js:283:19)
at bootstrapNodeJSCore (internal/bootstrap/node.js:622:3)
```


Debugging



Debugging

- Use the **debugger** keyword to stop and inspect your live code
- In the browser you need to keep the JavaScript console open
- In node.JS you need to call **node inspect**:

- Useful Doc:

http://www.w3schools.com/js/js_debugging.asp

<https://nodejs.org/api/debugger.html>

Hands On 6: Debugging



Reopen constant-error.js and add a debugger statement.

```
const fs = require('fs');  
const path = require('path');
```

```
debugger;
```

```
// Assign a new property to the fs object.  
fs.aNewProperty = 'some value';  
// Reassign the fs object.  
fs = 'a new life';
```

Hands On 6: Debugging



```
balistef@mzes072 MINGW64 ~/www/nodegame-workshop (master)
$ node inspect constant-error.js
< Debugger listening on ws://127.0.0.1:9229/4538a21a-c
< 002-4120-8dcd-19e93c2f2cff
< For help, see: https://nodejs.org/en/docs/inspector
< Debugger attached.
Break on start in file:///C:/Users/balistef/www/nodegame-workshop/constant-error.js:1
> 1 const fs = require('fs');
  2 const path = require('path');
  3
debug> |
```

To launch the debugger:
node debug or node inspect

n: next line

s: step into a function call

Repl: enter into Read-eval-print loop

NPM: Node Package Manager

<https://www.npmjs.com/>



Neutral Pumpkin Mews

npm Enterprise

Products

Solutions

Resources

Docs

Support

npm



Search packages

Search


Join




Log In

Build amazing things

Essential JavaScript development tools that help you go to market faster and build powerful applications using modern open source code.

The 11 Lines that Almost Broke the Internet

2 contributors 

18 lines (11 sloc) | 222 Bytes Raw Blame History   

```
1 module.exports = leftpad;
2
3 function leftpad (str, len, ch) {
4   str = String(str);
5
6   var i = -1;
7
8   if (!ch && ch !== 0) ch = ' ';
9
10  len = len - str.length;
11
12  while (++i < len) {
13    str = ch + str;
14  }
15
16  return str;
17 }
```

<https://www.sciencealert.com/how-a-programmer-almost-broke-the-internet-by-deleting-11-lines-of-code>

NPM: Node Package Manager



```
npm install one-liner-joke
```

- Creates a `node_modules/` folder inside the same directory.
- It contains the requested module and all its dependencies.
- We can now require it and use it in our programs.

```
const joker = require('one-liner-joker');
```

```
let randomJoke = joker.getRandomJoke();  
console.log(randomJoke);
```

Linting Tool: JSHint

What is the option `-g` doing?

```
npm install -g jshint
```

```
jshint constant-error.js --show-non-errors
```

- You might be interested in one of the plugins for editors (vim, emacs, sublime...)