

# **MeTaProGramming In JaVaScRiPT**

Cerize Santos

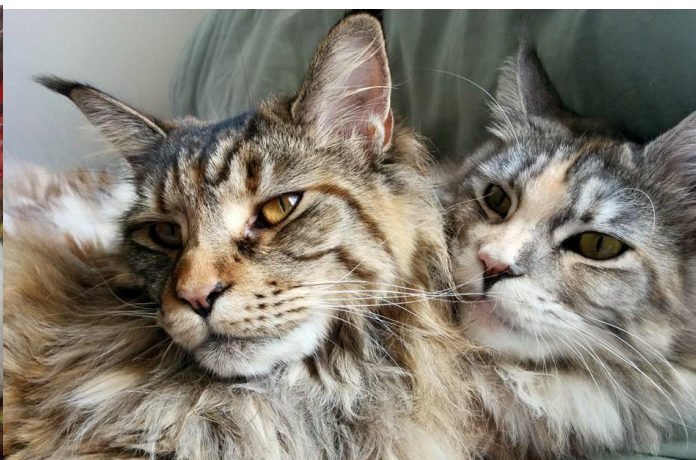
Women Who Code, April - 2017

# ABOUT me

Software developer at **Bananatag**

Work with **all things Javascript** - Node/React/Redux

Chemistry background

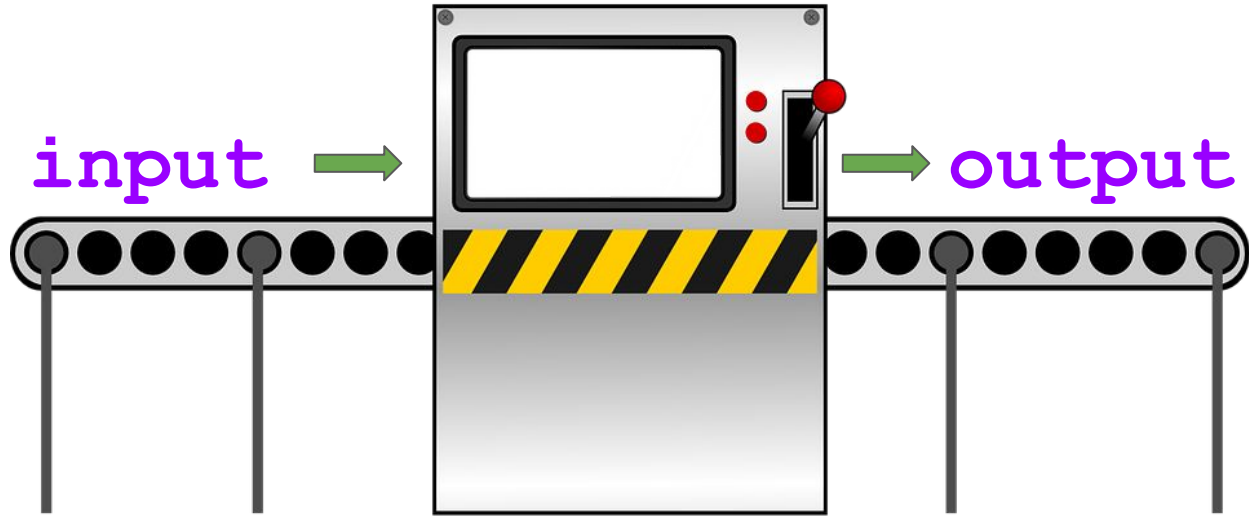


cats

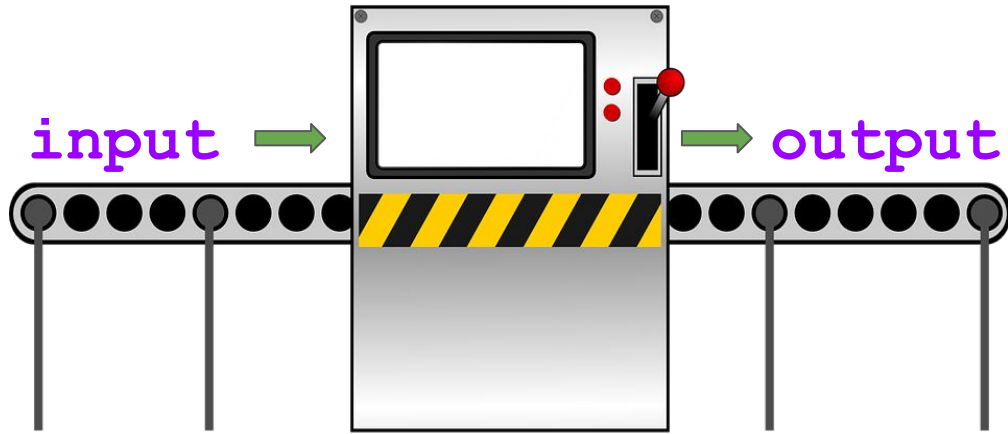
watercolour



# Programming - THE concept



# METAPROGRAMMING - THE CONCEPT



**Input:**

**Base:** user Input  
(string, array, browser  
event, etc)

**Meta:** Base level code

a program that can read, generate, analyse or transform other programs  
- or itself - while running

# examples

**Compiler: source code -> executable**

**IDEs**

**Scaffolding - Ruby on Rails made popular**

**ESLint**

**...Any code parser**

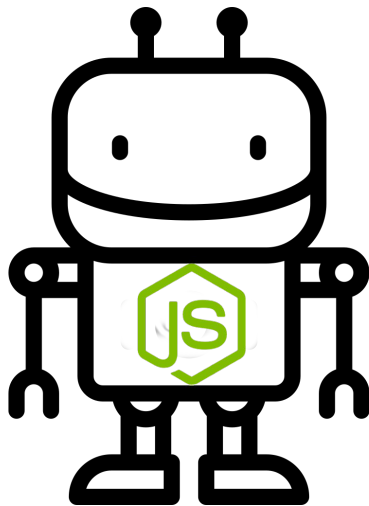
# ReFlection

metalanguage  $\neq$  language processed

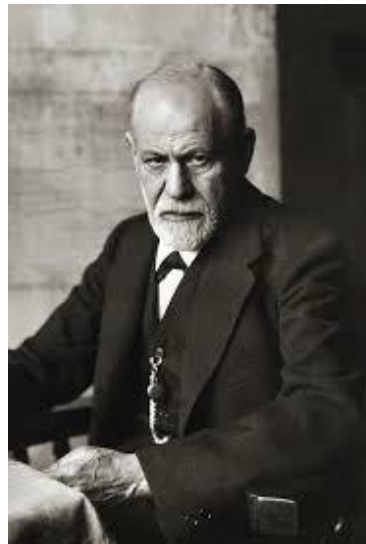


OK!

```
if (metalanguage === language processed) {  
    reflection or reflexivity.  
}
```



who am I ?



# REFLECTIVE metaprogramming\*

**Introspection:** program analyses its structure - read-only.

```
Object.keys()  
a.isPrototypeOf(b)
```

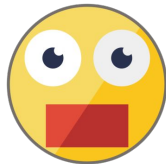
**Self-modification:** program changes its structure - write.

```
function moveProperty(source, propertyName, target) {  
    target[propertyName] = source[propertyName];  
    delete source[propertyName];  
}
```

**Intercession:** program redefines the semantics of some language operations .

Symbol.iterator

**Proxies**



# ES6 metaProgramming



# THE EVAL WITHIN



```
eval('2+2') // 4
```

```
const grayCat= { name: 'Molly', age: '3', favoriteToy: 'feather' };
```

```
console.log(eval('grayCat.name')); // Molly
```

**Proxy** - Standard built-in object

**new Proxy(target, handler)**

**target:** any object - even array, function, another proxy

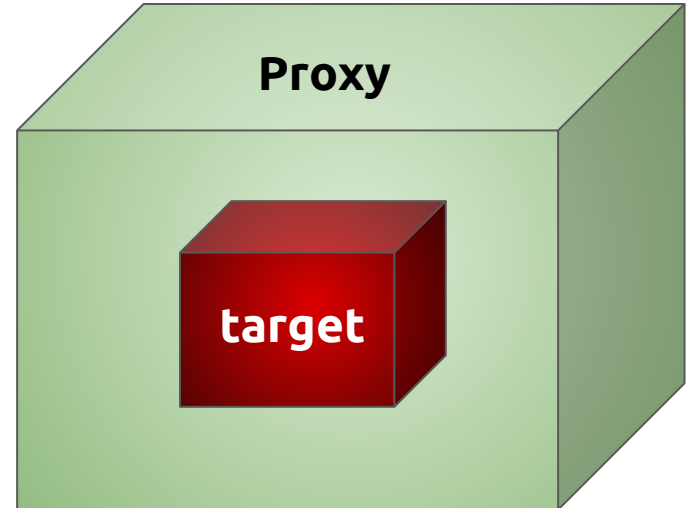
**handler:** {

has: {"A trap for the **in** operator"},

get: {"A trap for getting property values"}

}

-indistinguishable from an object



# Proxy example 1:

```
// Basic example

const target = {
  a: 1,
  b: undefined
};

const handler = {
  get: (target, propertyName) => {
    return propertyName in target ? target[propertyName] : 42;
  }
};

const p = new Proxy(target, handler);

console.log(p); // { a: 1, b: undefined }
console.log(p.a, p.b, p.c); // 1 undefined 42
console.log('a' in p, 'b' in p, 'c' in p); // true true false
```

## Proxy example 2:




# Meta Minesweeper

BOMBS: 8

GET A NEW BOARD

GO TO DEV MODE

					2	
	1		1			
0	0					
0	0		1	2	1	1
1	2			1	0	0
	4			1	0	0
				0	0	0



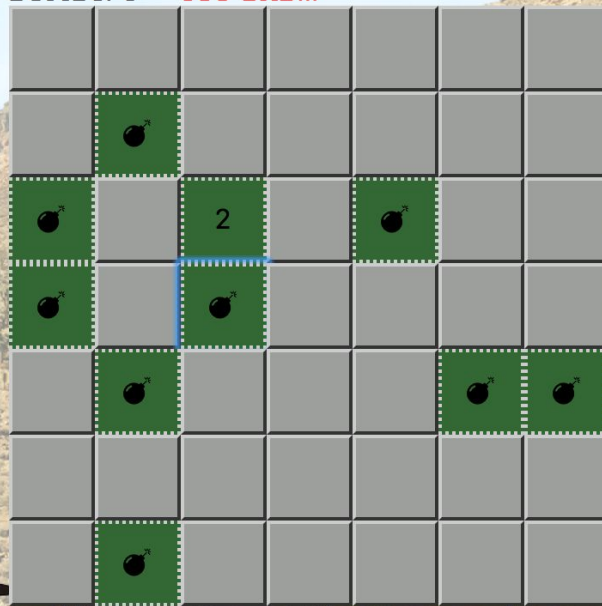


# Meta Minesweeper

GET A NEW BOARD

GO TO DEV MODE

BOMBS: 9    TOO BAD!!!



# Meta Minesweeper

## Dev Challenge

Which javascript built-in object allow you to intercept operations and implement custom behavior?

Cancel

Submit



```
const board = getInitialBoard(7);
```

```
class Game extends Component {
```

```
  render() {
```

```
    const { match } = this.props;
```

```
    const gameHandler = {
```

```
      set: (obj, prop, value) => {
```

```
        obj[prop] = value;
```

```
        // even if the new value is the same as previous, still rerenders
```

```
        // necessary because React won't see changes in deep nested props
```

```
        this.forceUpdate();
```

```
        return true;
```

```
      }
```

```
    };
```

```
    const proxyBoard = new Proxy(board, gameHandler);
```

```
    return (
```

```
      <div className="game">
```



```
class Board extends Component {
  _handleClick = (obj) => {
    // Flag placed: decrease the number of bombs
    if (obj.status === 'flag') {
      this.props.board.nBombs -= 1;
    }
    // Flag removed: decrease the number of bombs
    if (obj.status === 'closed') {
      this.props.board.state = 'evaluating click';
      this.props.board.nBombs += 1;
    }

    if (obj.status === 'open' && obj.face === 'bomb') {
      this._openAllBombs(this.props.board.squares);
      this.props.board.state = 'lost';
    } else {
      this.props.board.state = 'evaluating click';
    }

    if (this._checkVictory(this.props.board.squares)) {
      this.props.board.state = 'win';
    }
  }
}
```

```
const BoardRow = ({ row, board, mode, handleClick }) => {
  console.log('inside board, mode', mode);
  const squareHandler = {
    set: (obj, prop, value) => {
      obj[prop] = value;
      handleClick(obj);
      return true;
    }
  };
  return (
    <div className="board-row">
      {
        row.map((elem, index) => {
          const proxySquare = new Proxy(board.squares[elem], squareHandler);
          return (
            <div className="board-square">
              <Square
                key={index}
                square={proxySquare}
                mode={mode}
              />
            </div>
          );
        })
      }
    </div>
  );
}
```

```
class Square extends Component {
  _onRightClick = (e) => {
    e.preventDefault();
    if (this.props.square.status === 'open') {
      return;
    }
    this.props.square.status = this.props.square.status === 'closed' ? 'flag' : 'closed';
  };

  _onLeftClick = () => {
    // Triggers opening the question modal
    if (this.props.mode === 'dev') {
      this.props.square.status = 'question';
      return;
    }
    this.props.square.status = 'open';
  }

  _onCloseModal = () => {
    console.log('modal closed');
    this.props.square.status = 'quitQuestion';
  }

  _onSubmitAnswer = (e) => {
    e.preventDefault();
    const userAnswer = this.userAnswer.value;
    if (userAnswer === this.props.square.answer) {
      this.props.square.status = 'open';
      return;
    }
  }
}
```

# DISADVANTAGES

- Can be more costly to maintain
- *"Do not make a simple problem complex."*
- Proxies are not transpiled with Babel
- Proxies are only supported in latest versions of Chrome and Firefox



# Resources

- <https://www.slideshare.net/javierarilos/es6-metaprogramming-unleashed-49683123>
- [http://exploringjs.com/es6/ch\\_proxies.html](http://exploringjs.com/es6/ch_proxies.html)
- <https://github.com/getify/You-Dont-Know-JS/blob/master/es6%20%26%20beyond/ch7.md>

# THANK you!

Get in touch

[cerizesantos@gmail.com](mailto:cerizesantos@gmail.com)

Check it out the meta-minesweeper game

<https://github.com/cerize/minesweeper>