



Proxy e Reflect

Um proxy é capaz de **interceptar** diversos tipos de operações em um objeto alvo

A screenshot of a Mac OS X desktop environment showing a terminal window. The terminal window has a title bar "proxy_reflect_1.js — javascriptmasterclass". The main pane displays the following Node.js code:

```
JS proxy_reflect_1.js ×
1  function createArray() {
2    return {};
3  }
4  const languages = createArray();
5  languages[0] = "Python";
6  languages[1] = "Ruby";
7  languages[2] = "JavaScript";
8  console.log(languages);
9  console.log(languages.length);
10
```

The right pane of the terminal window shows the command-line interface output:

```
rodrigobranas:javascriptmasterclass $ node proxy_reflect/proxy_reflect_1.js
{ '0': 'Python', '1': 'Ruby', '2': 'JavaScript' }
undefined
rodrigobranas:javascriptmasterclass $ █
```

A screenshot of a macOS desktop environment. On the left is a code editor window titled "proxy_reflect_2.js — javascriptmasterclass". The code editor contains the following JavaScript code:

```
JS proxy_reflect_2.js x
1  function createArray() {
2      return new Proxy({}, {
3          });
4  }
5  const languages = createArray();
6  languages[0] = "Python";
7  languages[1] = "Ruby";
8  languages[2] = "JavaScript";
9  console.log(languages);
10 console.log(languages.length);
11
```

The code defines a function `createArray` that returns a proxy object. This proxy is used to create an array named `languages`. The array is populated with three elements: "Python", "Ruby", and "JavaScript". Finally, two logs are printed: the array itself and its length.

On the right is a terminal window titled "TERMINAL" with a tab labeled "1: bash". The terminal shows the command `node proxy_reflect/proxy_reflect_2.js` being run, followed by the output:

```
rodrigobranas:javascriptmasterclass $ node proxy_reflect/proxy_reflect_2.js
{ '0': 'Python', '1': 'Ruby', '2': 'JavaScript' }
undefined
rodrigobranas:javascriptmasterclass $
```

Existem métodos, chamados de trap, para diversos tipos de eventos relacionados a um objeto como:

- apply
- construct
- defineProperty
- deleteProperty
- get
- getOwnPropertyDescriptor
- getPrototypeOf
- has
- isExtensible
- ownKeys
- preventExtensions
- set
- setPrototypeOf

O método **set** é invocado quando uma propriedade é definida no objeto

A screenshot of a Mac OS X desktop environment showing a terminal window and a code editor. The terminal window is titled 'proxy_reflect_3.js — javascriptmasterclass' and contains the following text:

```
rodrigobranas:javascriptmasterclass $ node proxy_reflect/proxy_reflect_3.js
{}
undefined
rodrigobranas:javascriptmasterclass $
```

The code editor window shows a file named 'proxy_reflect_3.js' with the following content:

```
1  function createArray() {
2      return new Proxy({}, {
3          set(target, key, value) {
4              }
5      });
6  }
7  const languages = createArray();
8  languages[0] = "Python";
9  languages[1] = "Ruby";
10 languages[2] = "JavaScript";
11 console.log(languages);
12 console.log(languages.length);
13
```

A screenshot of a Mac OS X desktop environment. On the left is a code editor window titled "proxy_reflect_4.js — javascriptmasterclass". The code editor contains the following JavaScript code:

```
JS proxy_reflect_4.js x
proxy_reflect_4.js — javascriptmasterclass

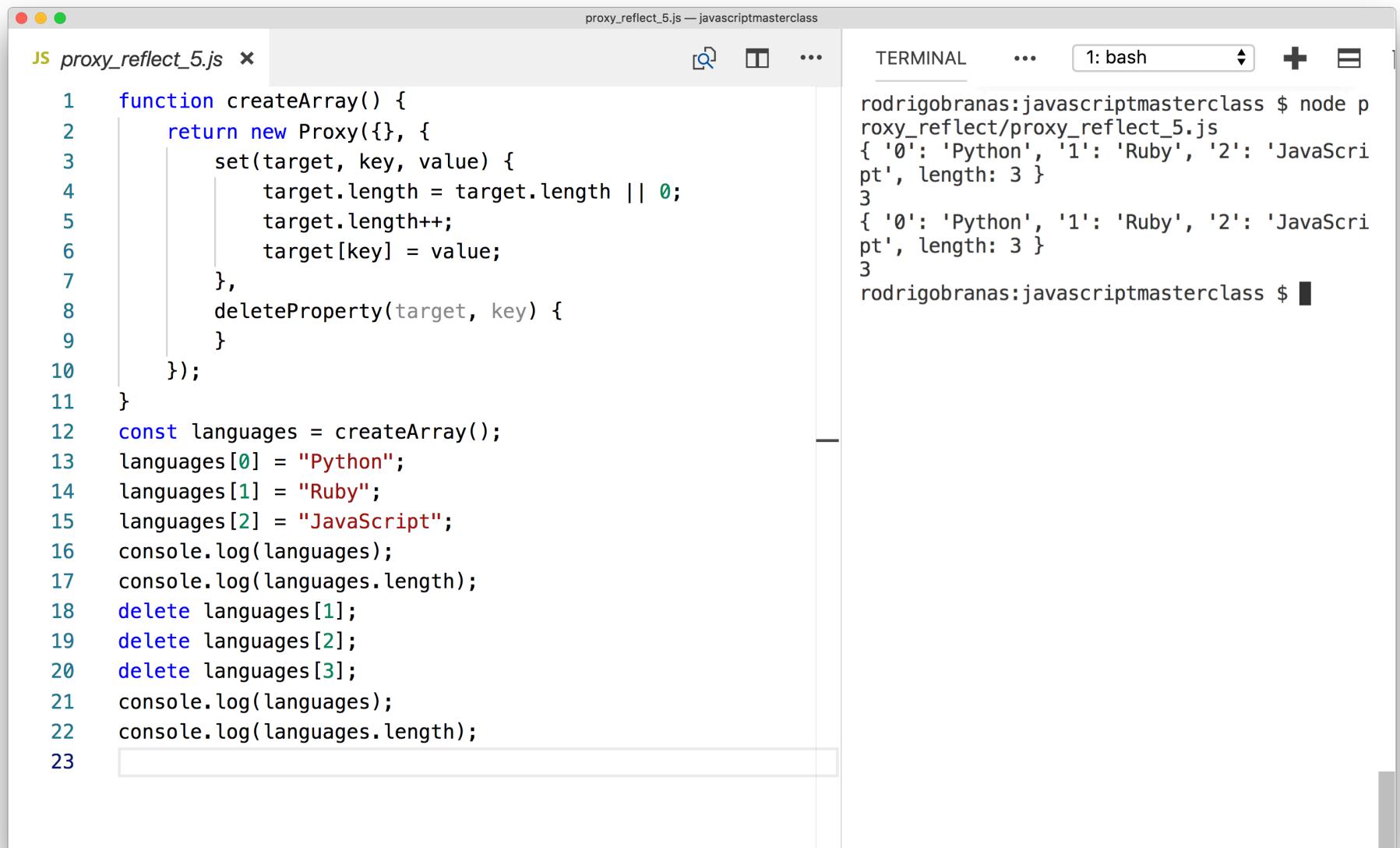
1  function createArray() {
2      return new Proxy({}, {
3          set(target, key, value) {
4              target.length = target.length || 0;
5              target.length++;
6              target[key] = value;
7          }
8      });
9  }
10 const languages = createArray();
11 languages[0] = "Python";
12 languages[1] = "Ruby";
13 languages[2] = "JavaScript";
14 console.log(languages);
15 console.log(languages.length);
16
```

The code defines a function `createArray` that returns a proxy object. This proxy has a `set` method which handles assignments to array elements. It ensures that the `length` property is updated correctly and that new elements are added to the array. The code then creates an array `languages` using this proxy, sets its elements to "Python", "Ruby", and "JavaScript", and logs both the array and its length to the console.

To the right of the code editor is a terminal window titled "TERMINAL" with a tab labeled "1: bash". The terminal shows the command `node proxy_reflect/proxy_reflect_4.js` being run, followed by the output:

```
rodrigobranas:javascriptmasterclass $ node proxy_reflect/proxy_reflect_4.js
{ '0': 'Python', '1': 'Ruby', '2': 'JavaScript', length: 3 }
3
rodrigobranas:javascriptmasterclass $
```

O método **deleteProperty** é invocado quando uma propriedade é deletada



A screenshot of a Mac OS X desktop environment showing a terminal window and a code editor. The terminal window is titled 'TERMINAL' and has a tab labeled '1: bash'. It displays the command 'node proxy_reflect/proxy_reflect_5.js' followed by its output. The code editor window is titled 'proxy_reflect_5.js — javascriptmasterclass'. It contains a JavaScript file with code demonstrating the use of the Proxy object to create an array-like object with custom behavior.

```
JS proxy_reflect_5.js x
proxy_reflect_5.js — javascriptmasterclass
1  function createArray() {
2      return new Proxy({}, {
3          set(target, key, value) {
4              target.length = target.length || 0;
5              target.length++;
6              target[key] = value;
7          },
8          deleteProperty(target, key) {
9          }
10     });
11 }
12 const languages = createArray();
13 languages[0] = "Python";
14 languages[1] = "Ruby";
15 languages[2] = "JavaScript";
16 console.log(languages);
17 console.log(languages.length);
18 delete languages[1];
19 delete languages[2];
20 delete languages[3];
21 console.log(languages);
22 console.log(languages.length);
23
```

```
rodrigobranas:javascriptmasterclass $ node proxy_reflect/proxy_reflect_5.js
{ '0': 'Python', '1': 'Ruby', '2': 'JavaScript', length: 3 }
3
{ '0': 'Python', '1': 'Ruby', '2': 'JavaScript', length: 3 }
3
rodrigobranas:javascriptmasterclass $
```

A screenshot of a Mac OS X desktop environment showing a terminal window and a code editor. The terminal window is titled 'proxy_reflect_6.js — javascriptmasterclass' and contains the following text:

```
rodrigobranas:javascriptmasterclass $ node proxy_reflect/proxy_reflect_6.js
{ '0': 'Python', '1': 'Ruby', '2': 'JavaScript', length: 3 }
3
{ '0': 'Python', length: 3 }
3
rodrigobranas:javascriptmasterclass $
```

The code editor window shows the file 'proxy_reflect_6.js' with the following content:

```
JS proxy_reflect_6.js x
proxy_reflect_6.js — javascriptmasterclass

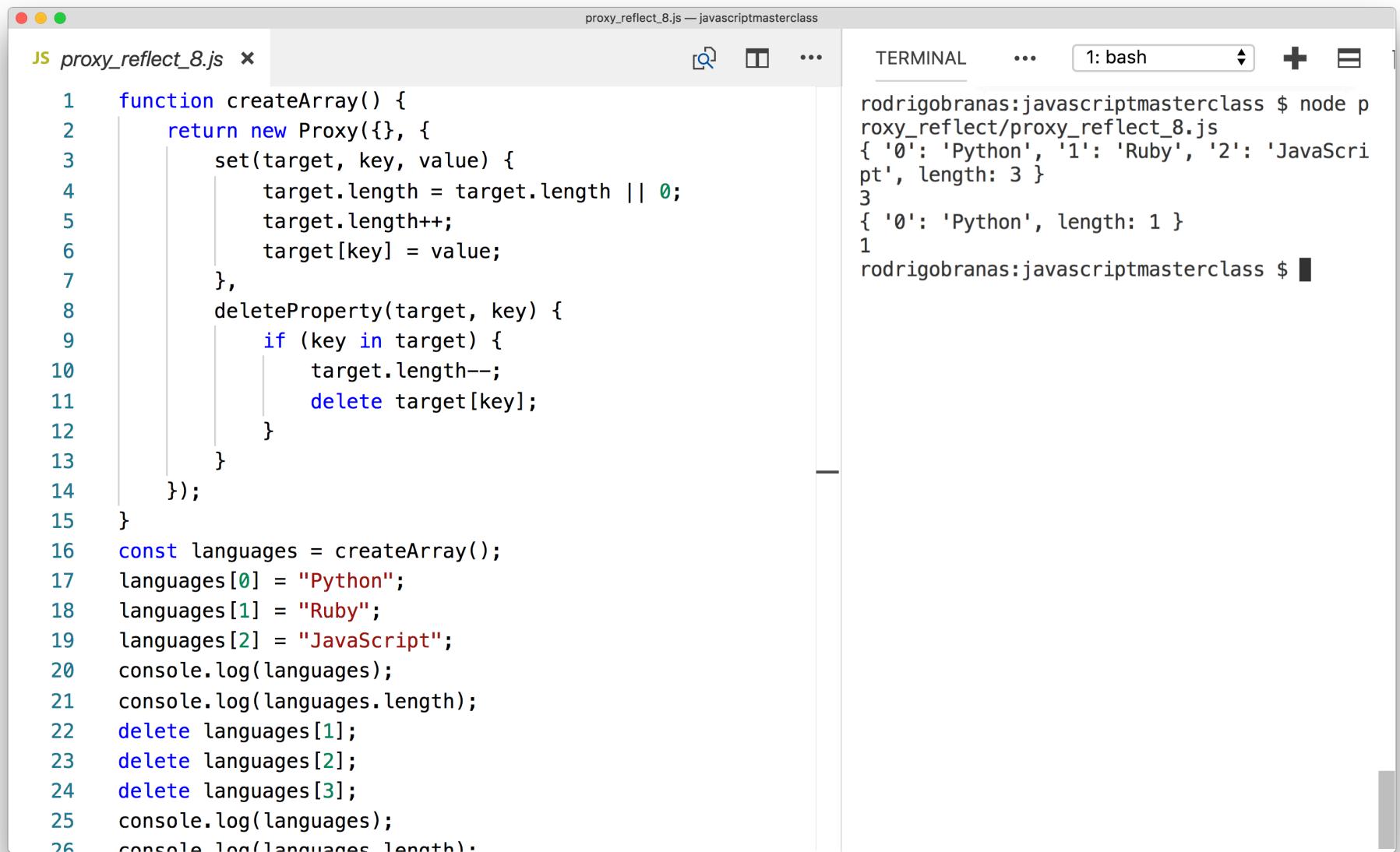
1  function createArray() {
2      return new Proxy({}, {
3          set(target, key, value) {
4              target.length = target.length || 0;
5              target.length++;
6              target[key] = value;
7          },
8          deleteProperty(target, key) {
9              delete target[key];
10         }
11     });
12 }
13 const languages = createArray();
14 languages[0] = "Python";
15 languages[1] = "Ruby";
16 languages[2] = "JavaScript";
17 console.log(languages);
18 console.log(languages.length);
19 delete languages[1];
20 delete languages[2];
21 delete languages[3];
22 console.log(languages);
23 console.log(languages.length);
24
```

A screenshot of a macOS desktop environment showing a terminal window and a code editor. The terminal window is titled 'proxy_reflect_7.js — javascriptmasterclass' and contains the following output:

```
rodrigobranas:javascriptmasterclass $ node proxy_reflect/proxy_reflect_7.js
{ '0': 'Python', '1': 'Ruby', '2': 'JavaScript', length: 3 }
3
{ '0': 'Python', length: 0 }
0
rodrigobranas:javascriptmasterclass $
```

The code editor window shows the file 'proxy_reflect_7.js' with the following content:

```
JS proxy_reflect_7.js x
proxy_reflect_7.js — javascriptmasterclass
1  function createArray() {
2      return new Proxy({}, {
3          set(target, key, value) {
4              target.length = target.length || 0;
5              target.length++;
6              target[key] = value;
7          },
8          deleteProperty(target, key) {
9              target.length--;
10             delete target[key];
11         }
12     });
13 }
14 const languages = createArray();
15 languages[0] = "Python";
16 languages[1] = "Ruby";
17 languages[2] = "JavaScript";
18 console.log(languages);
19 console.log(languages.length);
20 delete languages[1];
21 delete languages[2];
22 delete languages[3];
23 console.log(languages);
24 console.log(languages.length);
25
```



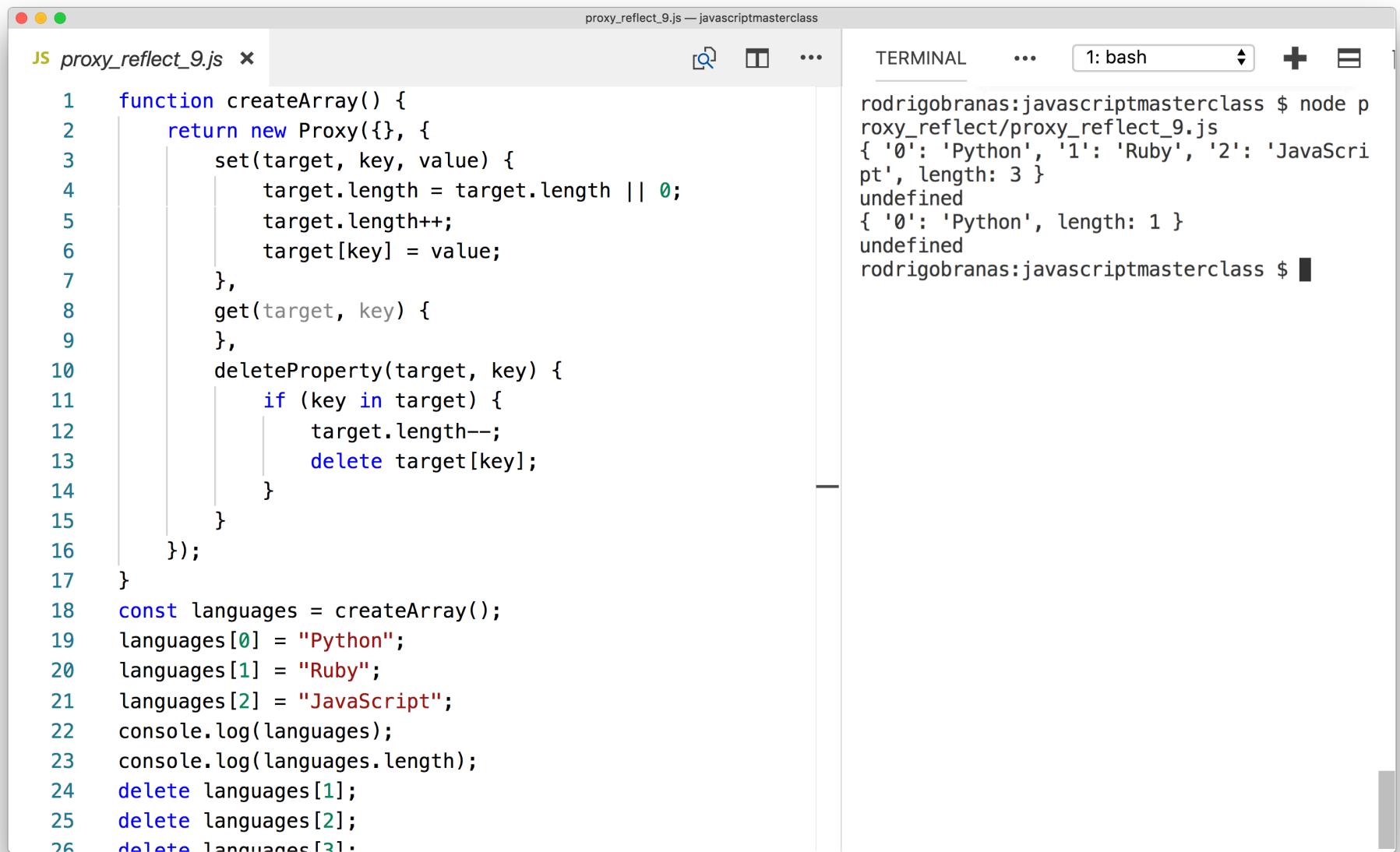
A screenshot of a Mac OS X desktop environment showing a terminal window and a code editor. The terminal window is titled 'TERMINAL' and has a tab labeled '1: bash'. It displays the command 'node proxy_reflect/proxy_reflect_8.js' followed by its output. The code editor window is titled 'proxy_reflect_8.js — javascriptmasterclass'. It contains a JavaScript file with code for creating an array-like object using a Proxy. The terminal output shows the creation of an array with three elements ('Python', 'Ruby', 'JavaScript'), a length of 3, and a delete operation that removes the second element, resulting in a length of 1.

```
rodrigobranas:javascriptmasterclass $ node proxy_reflect/proxy_reflect_8.js
{ '0': 'Python', '1': 'Ruby', '2': 'JavaScript', length: 3 }
3
{ '0': 'Python', length: 1 }
1
rodrigobranas:javascriptmasterclass $
```

```
JS proxy_reflect_8.js x
proxy_reflect_8.js — javascriptmasterclass
TERMINAL ... 1: bash + = 1

1  function createArray() {
2      return new Proxy({}, {
3          set(target, key, value) {
4              target.length = target.length || 0;
5              target.length++;
6              target[key] = value;
7          },
8          deleteProperty(target, key) {
9              if (key in target) {
10                  target.length--;
11                  delete target[key];
12              }
13          }
14      });
15  }
16  const languages = createArray();
17  languages[0] = "Python";
18  languages[1] = "Ruby";
19  languages[2] = "JavaScript";
20  console.log(languages);
21  console.log(languages.length);
22  delete languages[1];
23  delete languages[2];
24  delete languages[3];
25  console.log(languages);
26  console.log(languages.length);
```

O método **get** é invocado quando uma propriedade é acessada



A screenshot of a Mac OS X desktop environment showing a terminal window and a code editor. The terminal window is titled 'TERMINAL' and has a tab labeled '1: bash'. It displays the command 'node proxy_reflect/proxy_reflect_9.js' followed by its output. The code editor window is titled 'proxy_reflect_9.js — javascriptmasterclass'. It contains a JavaScript file with code for creating an array-like object using a Proxy. The terminal output shows the creation of an array with three elements ('Python', 'Ruby', 'JavaScript'), its length being 3, and then it is modified to have only one element ('Python') with length 1.

```
proxy_reflect_9.js — javascriptmasterclass
JS proxy_reflect_9.js ×
proxy_reflect_9.js — javascriptmasterclass
TERMINAL ... 1: bash + = 1

rodrigobranas:javascriptmasterclass $ node proxy_reflect/proxy_reflect_9.js
{ '0': 'Python', '1': 'Ruby', '2': 'JavaScript', length: 3 }
undefined
{ '0': 'Python', length: 1 }
undefined
rodrigobranas:javascriptmasterclass $
```

```
function createArray() {
  return new Proxy({}, {
    set(target, key, value) {
      target.length = target.length || 0;
      target.length++;
      target[key] = value;
    },
    get(target, key) {
    },
    deleteProperty(target, key) {
      if (key in target) {
        target.length--;
        delete target[key];
      }
    }
  });
}

const languages = createArray();
languages[0] = "Python";
languages[1] = "Ruby";
languages[2] = "JavaScript";
console.log(languages);
console.log(languages.length);
delete languages[1];
delete languages[2];
delete languages[2];
```

A screenshot of a terminal window titled "proxy_reflect_10.js — javascriptmasterclass". The terminal shows the output of running the JavaScript file "proxy_reflect_10.js" with the command "node proxy_reflect/proxy_reflect_10.js". The output displays the creation of an array-like object with properties at indices 0, 1, and 2, and a length property. It also shows the deletion of the first element and the subsequent decrease in length.

```
JS proxy_reflect_10.js x
proxy_reflect_10.js — javascriptmasterclass
TERMINAL ... 1: bash + = 1

rodrigobranas:javascriptmasterclass $ node proxy_reflect/proxy_reflect_10.js
{ '0': 'Python', '1': 'Ruby', '2': 'JavaScript', length: 3 }
3
{ '0': 'Python', length: 1 }
1
rodrigobranas:javascriptmasterclass $
```

```
1 function createArray() {
2     return new Proxy({}, {
3         set(target, key, value) {
4             target.length = target.length || 0;
5             target.length++;
6             target[key] = value;
7         },
8         get(target, key) {
9             return target[key];
10        },
11        deleteProperty(target, key) {
12            if (key in target) {
13                target.length--;
14                delete target[key];
15            }
16        }
17    });
18}
19const languages = createArray();
20languages[0] = "Python";
21languages[1] = "Ruby";
22languages[2] = "JavaScript";
23console.log(languages);
24console.log(languages.length);
25delete languages[1];
26delete languages[2].
```

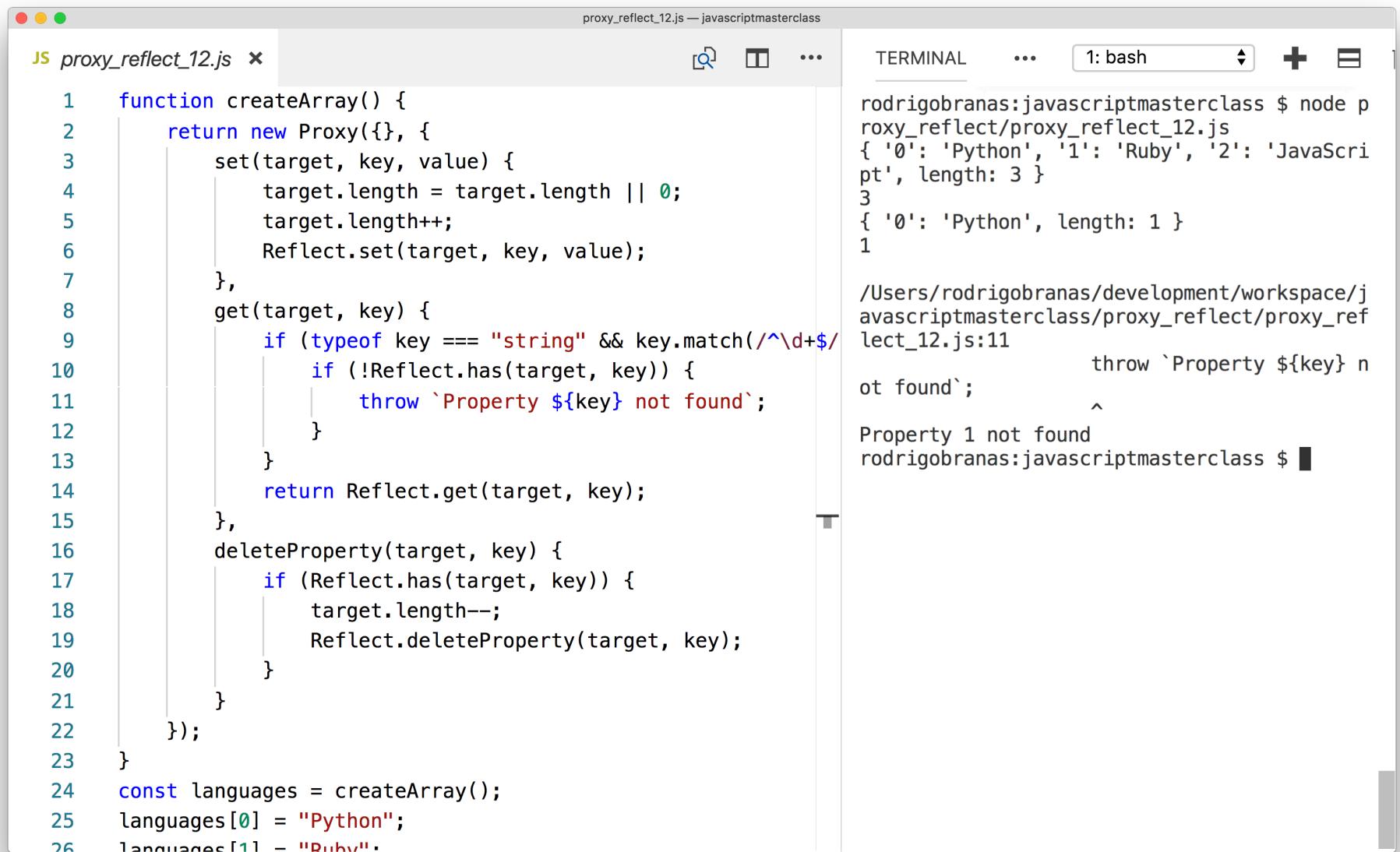
A screenshot of a macOS terminal window titled "proxy_reflect_11.js — javascriptmasterclass". The terminal shows the output of running the script, which creates an array-like object with properties at indices 0, 1, and 2, and then attempts to access a non-existent property "1" resulting in an error.

```
JS proxy_reflect_11.js x
proxy_reflect_11.js — javascriptmasterclass
TERMINAL ... 1: bash + = 1
rodrigobranas:javascriptmasterclass $ node proxy_reflect/proxy_reflect_11.js
{ '0': 'Python', '1': 'Ruby', '2': 'JavaScript', length: 3 }
3
{ '0': 'Python', length: 1 }
1

/Users/rodrigobranas/development/workspace/javascriptmasterclass/proxy_reflect/proxy_reflect_11.js:11
          throw `Property ${key} not found`;
                                         ^
Property 1 not found
rodrigobranas:javascriptmasterclass $
```

```
1  function createArray() {
2      return new Proxy({}, {
3          set(target, key, value) {
4              target.length = target.length || 0;
5              target.length++;
6              target[key] = value;
7          },
8          get(target, key) {
9              if (typeof key === "string" && key.match(/^\d+$/))
10                  if (!(key in target)) {
11                      throw `Property ${key} not found`;
12                  }
13              }
14              return target[key];
15          },
16          deleteProperty(target, key) {
17              if (key in target) {
18                  target.length--;
19                  delete target[key];
20              }
21          }
22      });
23  }
24  const languages = createArray();
25  languages[0] = "Python";
26  languages[1] = "Ruby";
```

A Reflect API tem os mesmos métodos que existem no Proxy, permitindo a execução de diversos tipos de operações no objeto alvo



A screenshot of a Mac OS X desktop environment showing a terminal window and a code editor. The terminal window is titled 'TERMINAL' and has a tab labeled '1: bash'. It displays the command 'node proxy_reflect/proxy_reflect_12.js' followed by its output. The code editor window is titled 'proxy_reflect_12.js — javascriptmasterclass'. It contains a JavaScript file with code for creating an array-like object using the Proxy API.

```
JS proxy_reflect_12.js x
proxy_reflect_12.js — javascriptmasterclass
1  function createArray() {
2      return new Proxy({}, {
3          set(target, key, value) {
4              target.length = target.length || 0;
5              target.length++;
6              Reflect.set(target, key, value);
7          },
8          get(target, key) {
9              if (typeof key === "string" && key.match(/^\d+$/))
10                 if (!Reflect.has(target, key)) {
11                     throw `Property ${key} not found`;
12                 }
13             }
14             return Reflect.get(target, key);
15         },
16         deleteProperty(target, key) {
17             if (Reflect.has(target, key)) {
18                 target.length--;
19                 Reflect.deleteProperty(target, key);
20             }
21         }
22     });
23 }
24 const languages = createArray();
25 languages[0] = "Python";
26 languages[1] = "Ruby".
```

```
TERMINAL ... 1: bash + =
rodrigobranas:javascriptmasterclass $ node proxy_reflect/proxy_reflect_12.js
{ '0': 'Python', '1': 'Ruby', '2': 'JavaScript', length: 3 }
3
{ '0': 'Python', length: 1 }
1

/Users/rodrigobranas/development/workspace/javascriptmasterclass/proxy_reflect/proxy_reflect_12.js:11
                throw `Property ${key} not found`;
                           ^
Property 1 not found
rodrigobranas:javascriptmasterclass $
```