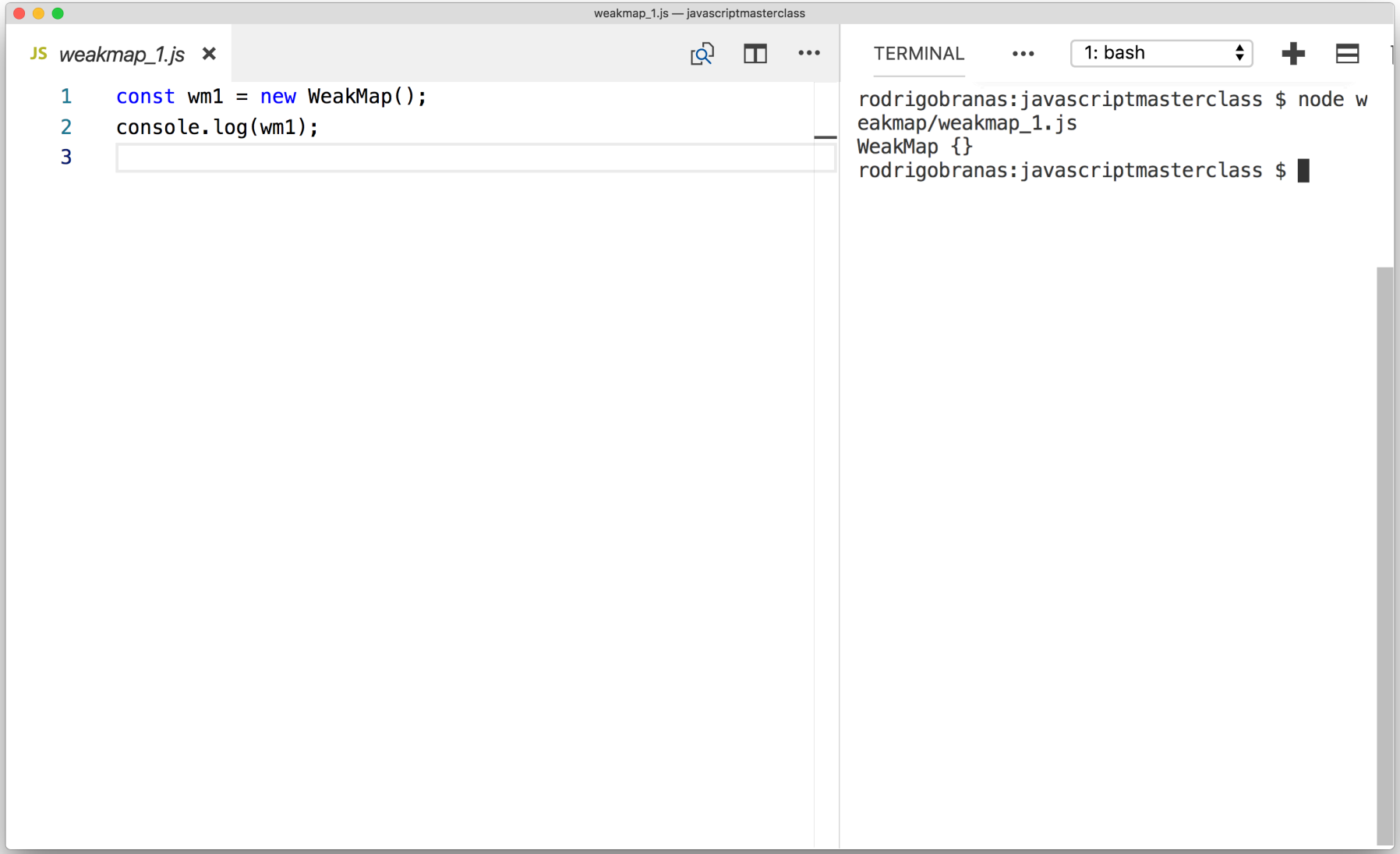
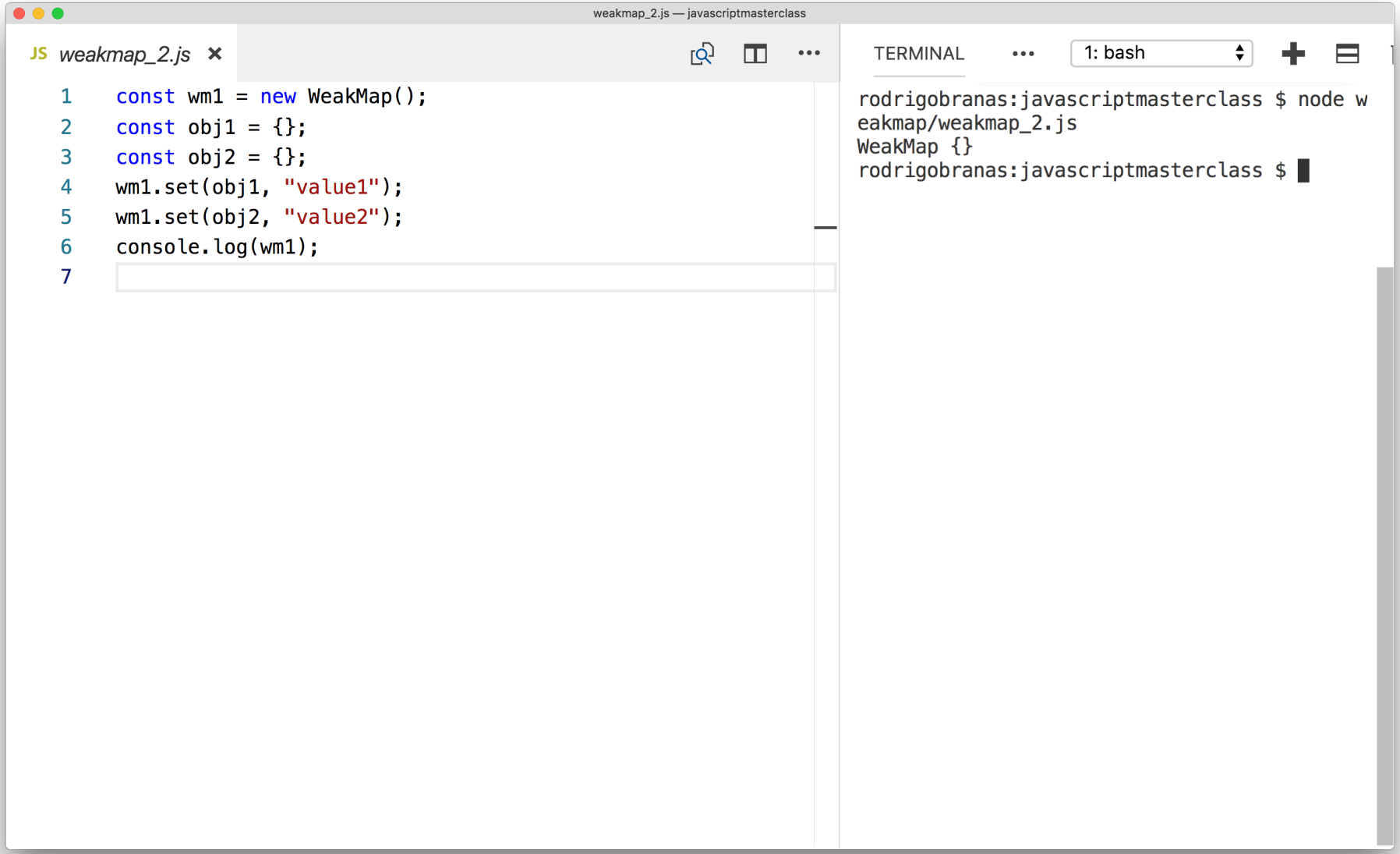
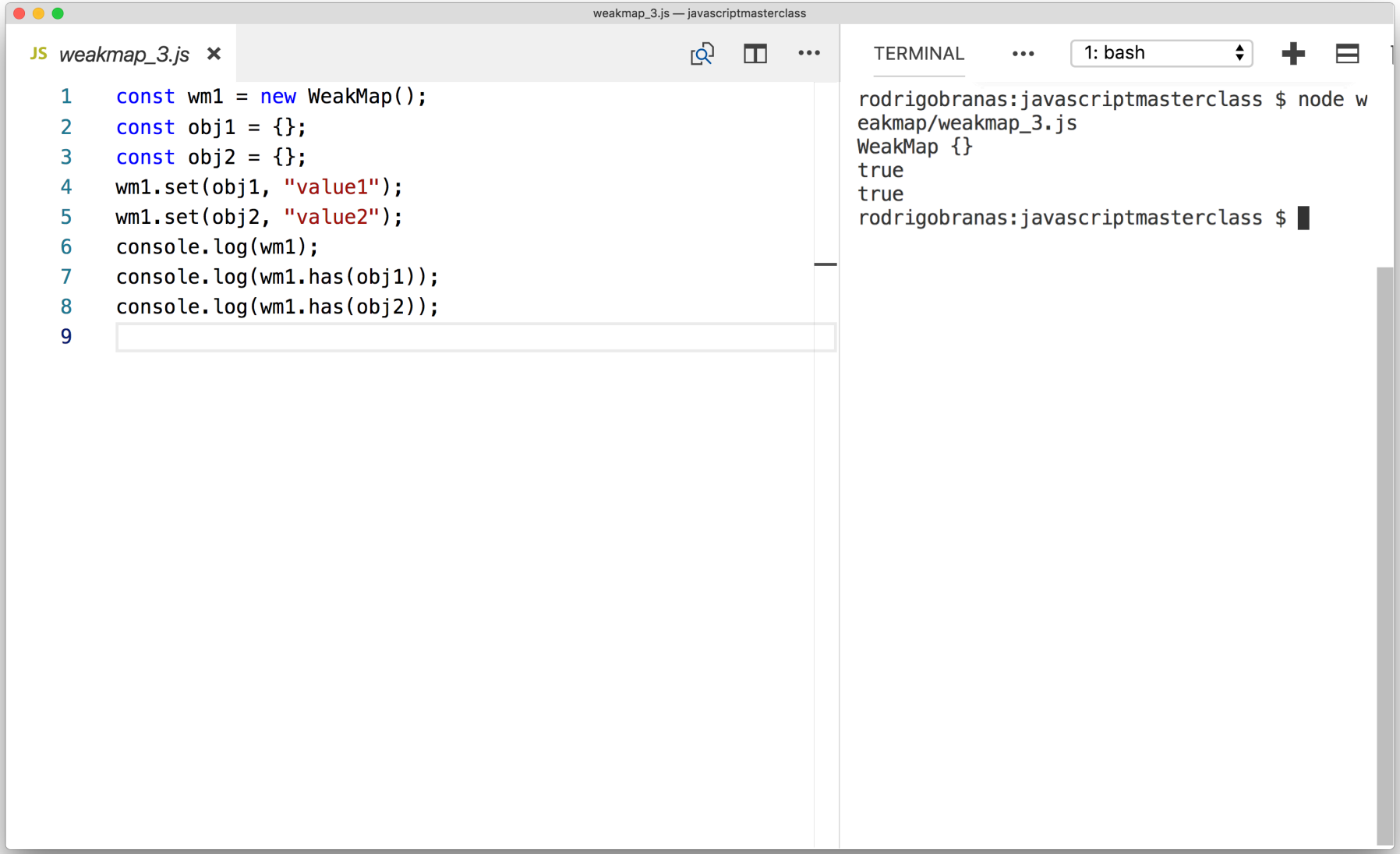
WeakMap

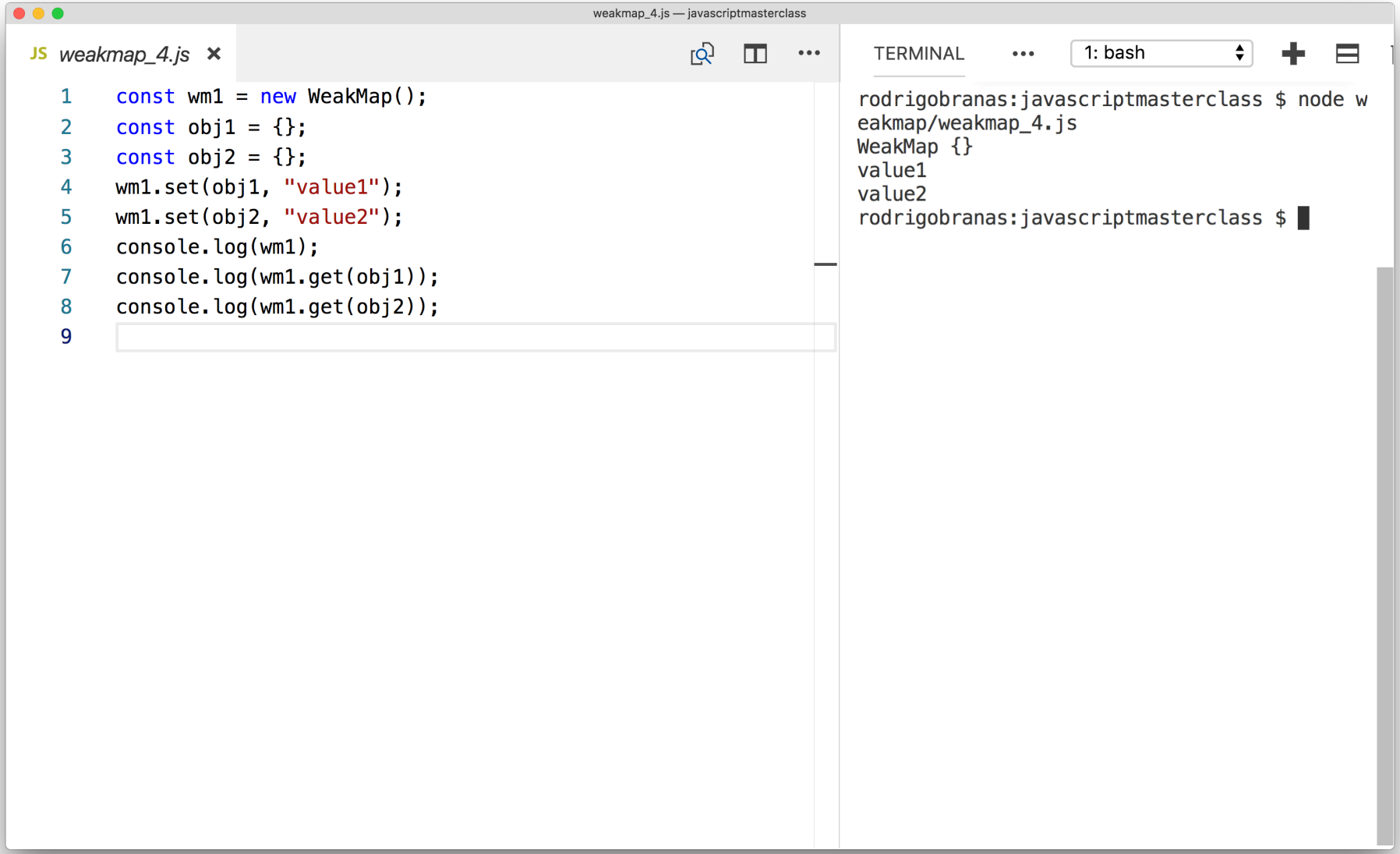
WeakMap é um objeto, similar ao Map, que permite apenas chaves do tipo Object e mantém as referências de forma fraca, sendo volátil e não iterável

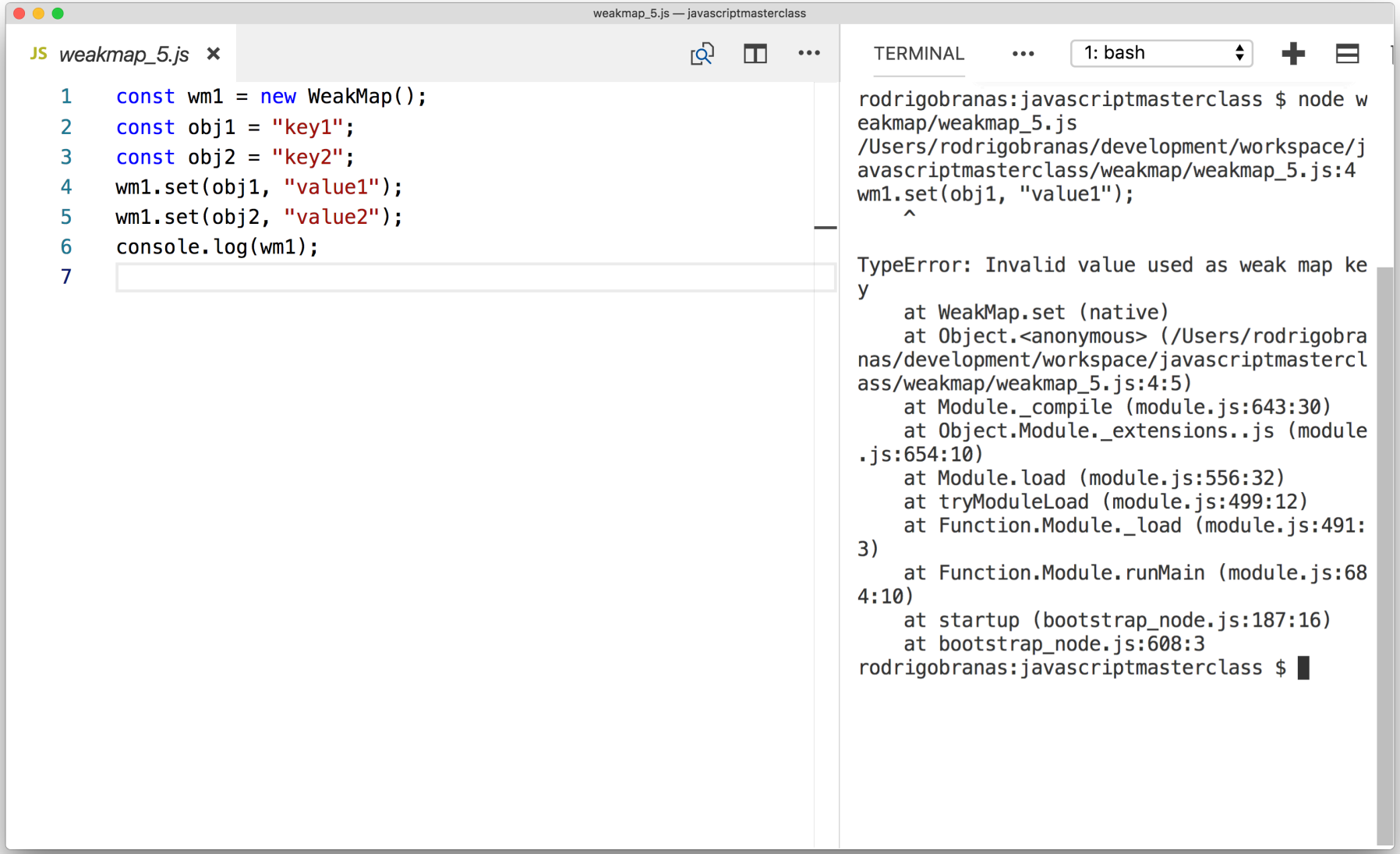

weakmap_1.js — javascriptmasterclass
JS weakmap_1.js
1 const wm1 = new WeakMap();
2 console.log(wm1);
3
TERMINAL
1: bash
rodrigobranas:javascriptmasterclass $ node w
eakmap/weakmap_1.js
WeakMap {}
rodrigobranas:javascriptmasterclass $

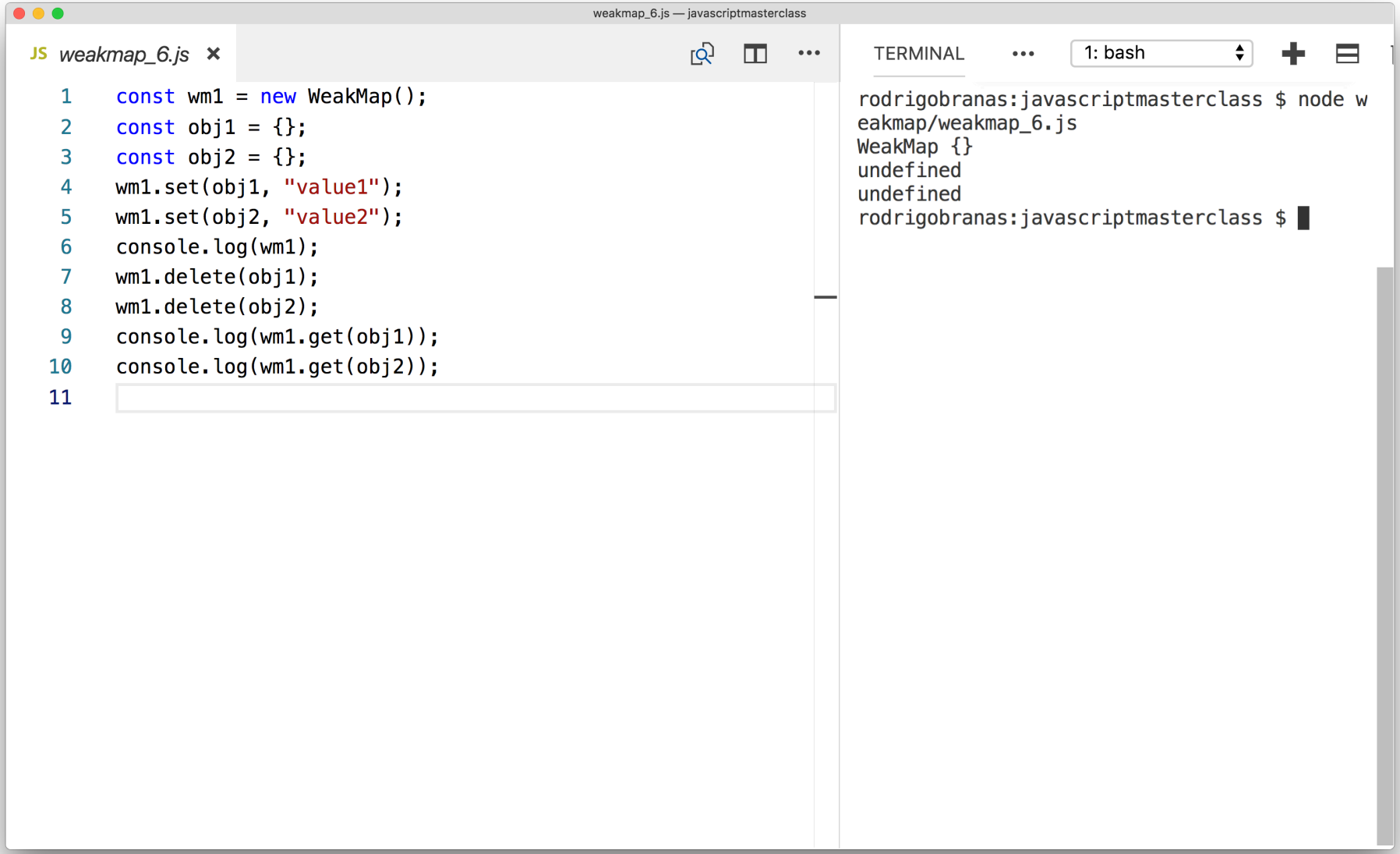
- set: Adiciona um par de chave e valor
- has: Retorna true se a chave existir
- get: Retorna o valor de uma determinada chave
- delete: Remove um par de chave e valor

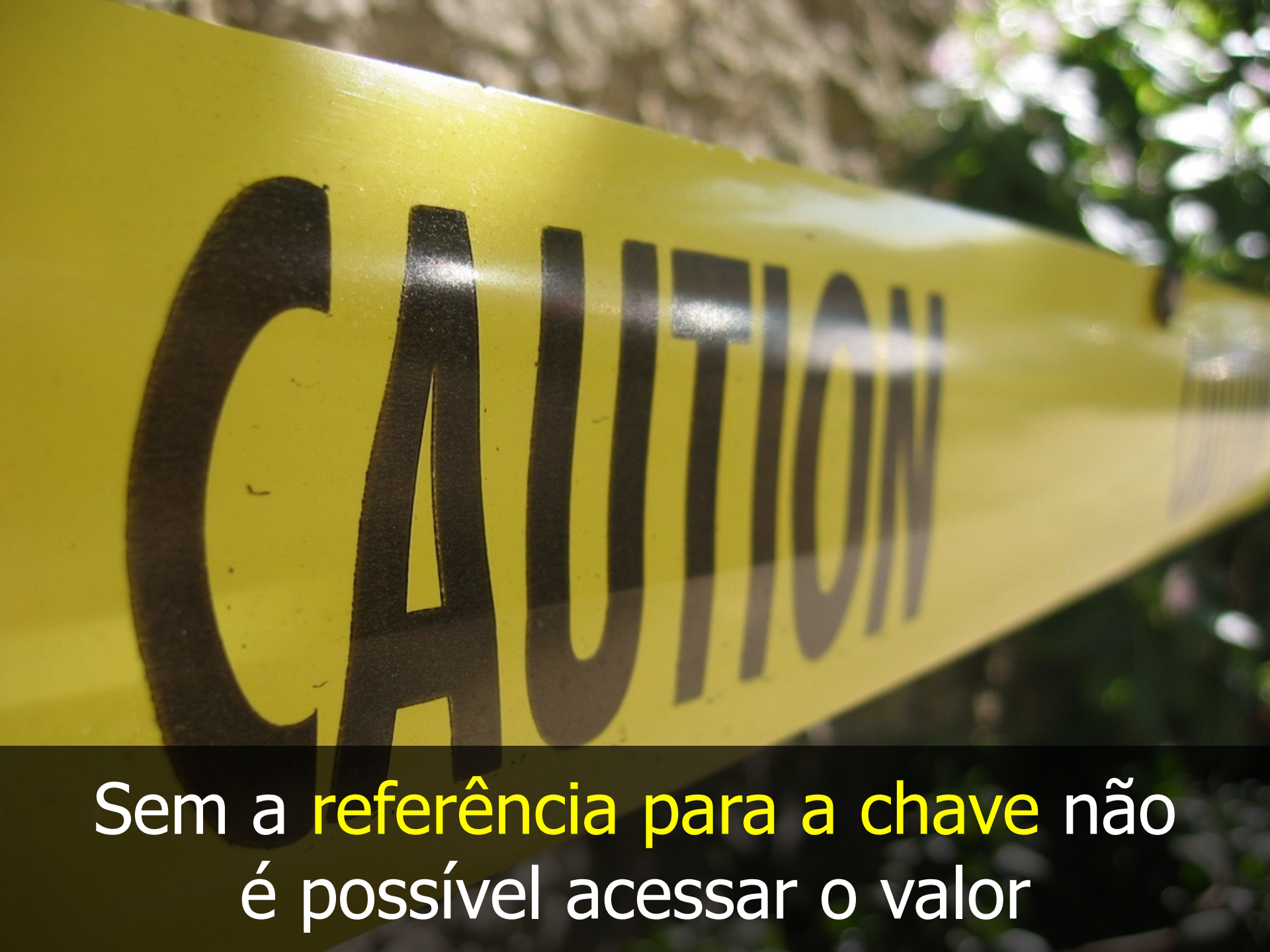

weakmap_2.js — javascriptmasterclass
JS weakmap_2.js
1 const wm1 = new WeakMap();
2 const obj1 = {};
3 const obj2 = {};
4 wm1.set(obj1, "value1");
5 wm1.set(obj2, "value2");
6 console.log(wm1);
7
TERMINAL
1: bash
rodrigobranas:javascriptmasterclass $ node w
eakmap/weakmap_2.js
WeakMap {}
rodrigobranas:javascriptmasterclass $


weakmap_3.js — javascriptmasterclass
JS weakmap_3.js
1 const wm1 = new WeakMap();
2 const obj1 = {};
3 const obj2 = {};
4 wm1.set(obj1, "value1");
5 wm1.set(obj2, "value2");
6 console.log(wm1);
7 console.log(wm1.has(obj1));
8 console.log(wm1.has(obj2));
9
TERMINAL
1: bash
rodrigobranas:javascriptmasterclass $ node w
eakmap/weakmap_3.js
WeakMap {}
true
true
rodrigobranas:javascriptmasterclass $

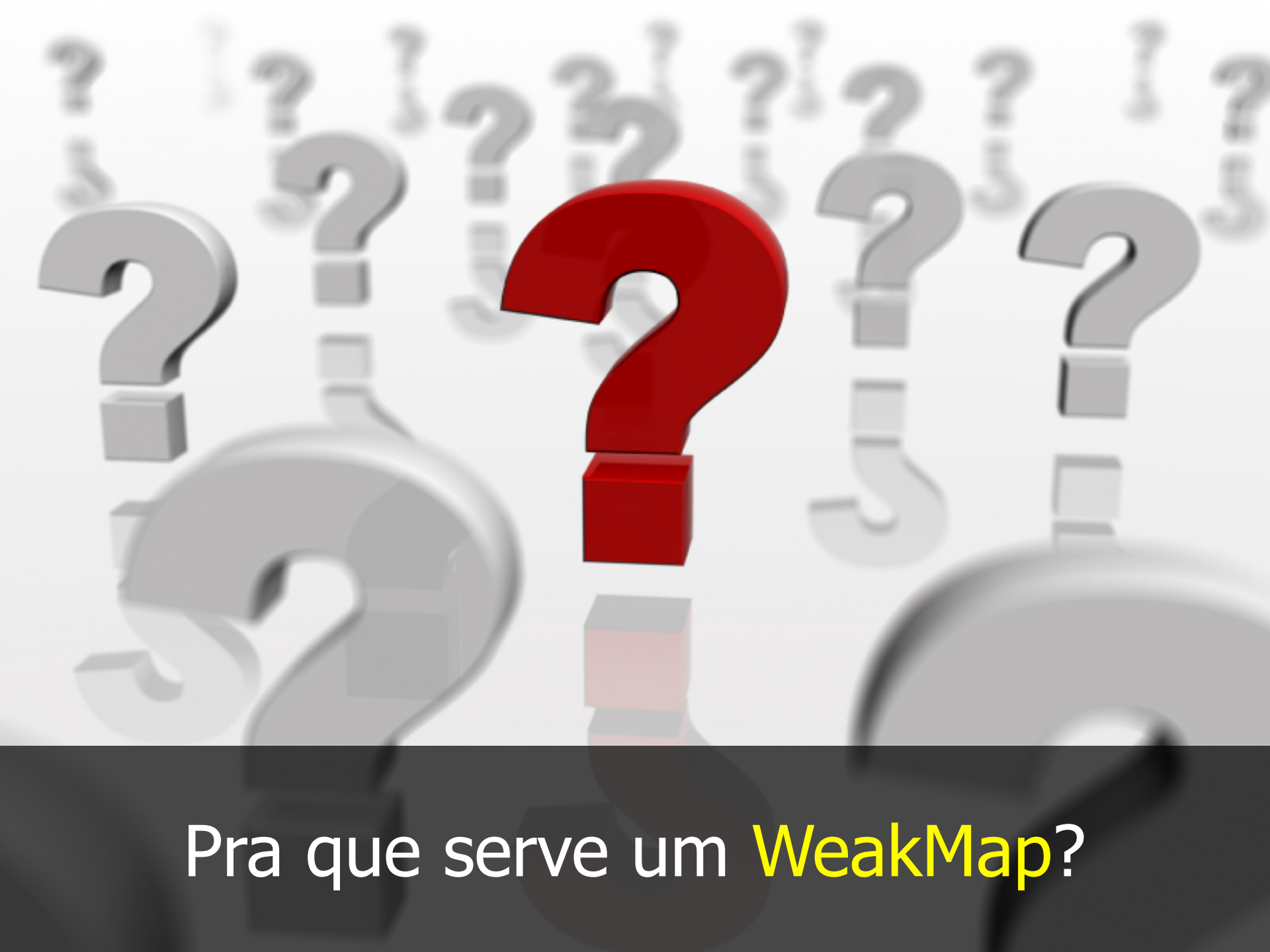

weakmap_4.js — javascriptmasterclass
JS weakmap_4.js
1 const wm1 = new WeakMap();
2 const obj1 = {};
3 const obj2 = {};
4 wm1.set(obj1, "value1");
5 wm1.set(obj2, "value2");
6 console.log(wm1);
7 console.log(wm1.get(obj1));
8 console.log(wm1.get(obj2));
9
TERMINAL
1: bash
rodrigobranas:javascriptmasterclass $ node w
eakmap/weakmap_4.js
WeakMap {}
value1
value2
rodrigobranas:javascriptmasterclass $

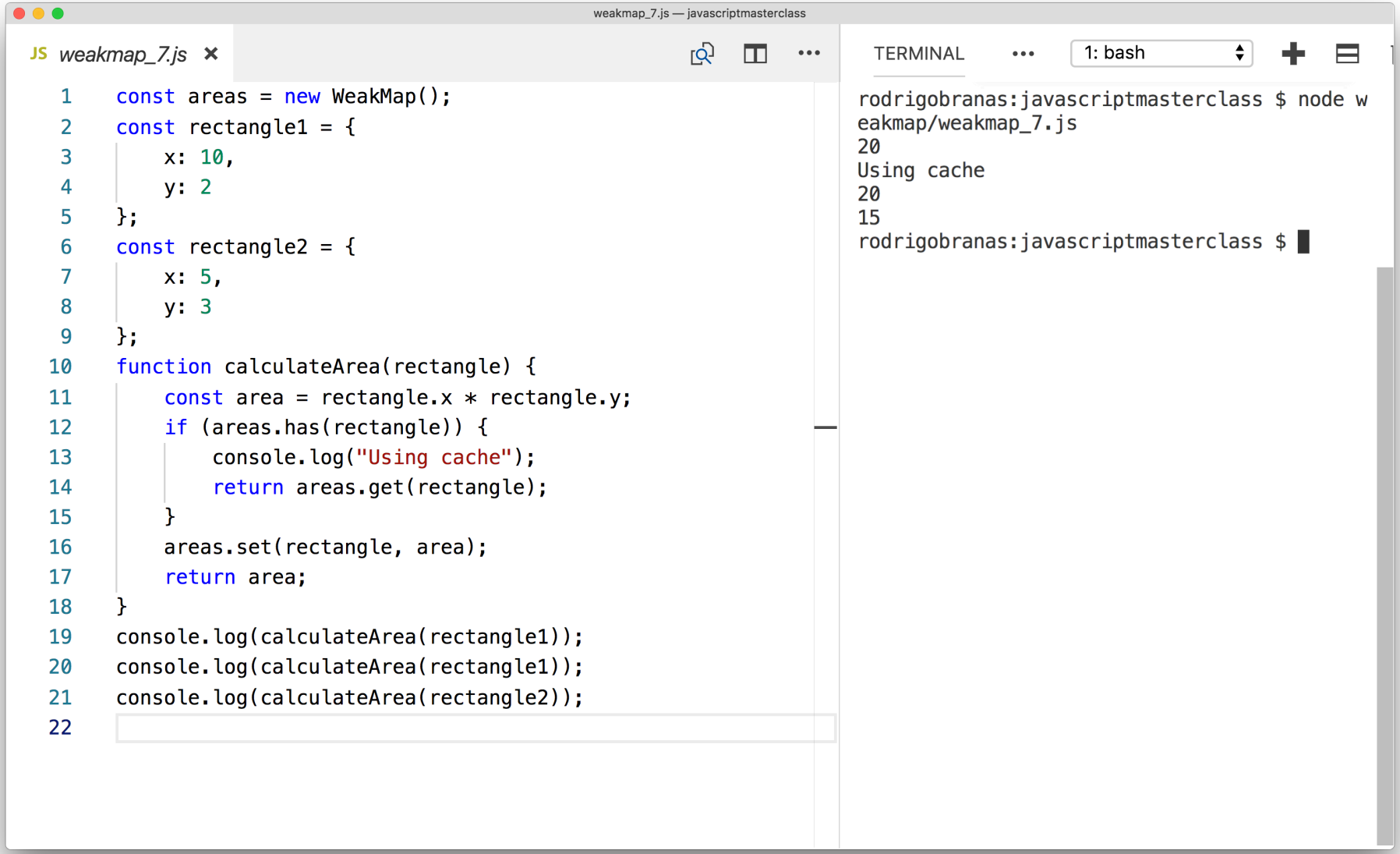

weakmap_5.js — javascriptmasterclass
JS weakmap_5.js
1 const wm1 = new WeakMap();
2 const obj1 = "key1";
3 const obj2 = "key2";
4 wm1.set(obj1, "value1");
5 wm1.set(obj2, "value2");
6 console.log(wm1);
7
TERMINAL
1: bash
rodrigobranas:javascriptmasterclass $ node weakmap/weakmap_5.js
/Users/rodrigobranas/development/workspace/javascriptmasterclass/weakmap/weakmap_5.js:4
wm1.set(obj1, "value1");
^
TypeError: Invalid value used as weak map key
at WeakMap.set (native)
at Object.<anonymous> (/Users/rodrigobranas/development/workspace/javascriptmasterclass/weakmap/weakmap_5.js:4:5)
at Module._compile (module.js:643:30)
at Object.Module._extensions..js (module.js:654:10)
at Module.load (module.js:556:32)
at tryModuleLoad (module.js:499:12)
at Function.Module._load (module.js:491:3)
at Function.Module.runMain (module.js:684:10)
at startup (bootstrap_node.js:187:16)
at bootstrap_node.js:608:3
rodrigobranas:javascriptmasterclass $


weakmap_6.js — javascriptmasterclass
JS weakmap_6.js
1 const wm1 = new WeakMap();
2 const obj1 = {};
3 const obj2 = {};
4 wm1.set(obj1, "value1");
5 wm1.set(obj2, "value2");
6 console.log(wm1);
7 wm1.delete(obj1);
8 wm1.delete(obj2);
9 console.log(wm1.get(obj1));
10 console.log(wm1.get(obj2));
11
TERMINAL
1: bash
rodrigobranas:javascriptmasterclass $ node w
eakmap/weakmap_6.js
WeakMap {}
undefined
undefined
rodrigobranas:javascriptmasterclass $



Sem a referência para a chave não é possível acessar o valor


Pra que serve um WeakMap?


weakmap_7.js — javascriptmasterclass
JS weakmap_7.js
TERMINAL
1: bash
const areas = new WeakMap();
const rectangle1 = {
x: 10,
y: 2
};
const rectangle2 = {
x: 5,
y: 3
};
function calculateArea(rectangle) {
const area = rectangle.x * rectangle.y;
if (areas.has(rectangle)) {
console.log("Using cache");
return areas.get(rectangle);
}
areas.set(rectangle, area);
return area;
}
console.log(calculateArea(rectangle1));
console.log(calculateArea(rectangle1));
console.log(calculateArea(rectangle2));
rodrigobranas:javascriptmasterclass $ node w
eakmap/weakmap_7.js
20
Using cache
20
15
rodrigobranas:javascriptmasterclass $