ES2015 中的箭头函数和词法 this

发表于 [2017年02月14日](http://www.css88.com/archives/6959) by [愚人码头](http://www.css88.com/archives/author/admin) 被浏览 1,672 次 分享到：

**0**

**小编推荐：**[掘金](http://gold.xitu.io/welcome/frontend/?utm_source=css88&utm_medium=word&utm_content=ganhuo&utm_campaign=q3_personal)是一个高质量的技术社区，从 ECMAScript 6 到 Vue.js，性能优化到开源类库，让你不错过前端开发的每一个技术干货。各大应用市场搜索「掘金」即可下载APP，技术干货尽在掌握..

箭头函数是使用=>语法对函数定义的简写。它们在语法上类似于 C#，Java 8 和 CoffeeScript 中的相关特性。它们支持表达式（Expression bodies）和函数体（Statement bodies）。与函数不同，箭头函数与其上下文代码共享相同的词法this（愚人码头注：箭头函数并没有自己的this，它的this是派生而来的，根据“词法作用域”派生而来）。如果箭头函数在另一个函数体内，它共享其父函数的 arguments 变量。

JavaScript 代码:

1. // 使用表达式（Expression bodies）
2. var odds = evens.map(v => v + 1);
3. var nums = evens.map((v, i) => v + i);
5. // 使用函数体（Statement bodies）
6. nums.forEach(v => {
7. if (v % 5 === 0)
8. fives.push(v);
9. });
11. // 词法`this`
12. var bob = {
13. \_name: "Bob",
14. \_friends: [],
15. printFriends() {
16. this.\_friends.forEach(f =>
17. console.log(this.\_name + " knows " + f));
18. }
19. };
21. // 词法 arguments
22. function square() {
23. let example = () => {
24. let numbers = [];
25. for (let number of arguments) {
26. numbers.push(number \* number);
27. }
29. return numbers;
30. };
32. return example();
33. }
35. square(2, 4, 7.5, 8, 11.5, 21); // returns: [4, 16, 56.25, 64, 132.25, 441]

更多信息: [MDN Arrow Functions](https://developer.mozilla.org/en/docs/Web/JavaScript/Reference/Functions/Arrow_functions)

愚人码头注：

用Babel转译一下上面的代码，结果：

JavaScript 代码:

1. "use strict";
3. // Expression bodies
4. var odds = evens.map(function (v) {
5. return v + 1;
6. });
7. var nums = evens.map(function (v, i) {
8. return v + i;
9. });
11. // Statement bodies
12. nums.forEach(function (v) {
13. if (v % 5 === 0) fives.push(v);
14. });
16. // Lexical this
17. var bob = {
18. \_name: "Bob",
19. \_friends: [],
20. printFriends: function printFriends() {
21. var \_this = this;
23. this.\_friends.forEach(function (f) {
24. return console.log(\_this.\_name + " knows " + f);
25. });
26. }
27. };
29. // Lexical arguments
30. function square() {
31. var \_arguments = arguments;
33. var example = function example() {
34. var numbers = [];
35. var \_iteratorNormalCompletion = true;
36. var \_didIteratorError = false;
37. var \_iteratorError = undefined;
39. try {
40. for (var \_iterator = \_arguments[Symbol.iterator](), \_step; !(\_iteratorNormalCompletion = (\_step = \_iterator.next()).done); \_iteratorNormalCompletion = true) {
41. var number = \_step.value;
43. numbers.push(number \* number);
44. }
45. } catch (err) {
46. \_didIteratorError = true;
47. \_iteratorError = err;
48. } finally {
49. try {
50. if (!\_iteratorNormalCompletion && \_iterator.return) {
51. \_iterator.return();
52. }
53. } finally {
54. if (\_didIteratorError) {
55. throw \_iteratorError;
56. }
57. }
58. }
60. return numbers;
61. };
63. return example();
64. }
66. square(2, 4, 7.5, 8, 11.5, 21); // returns: [4, 16, 56.25, 64, 132.25, 441]

值得注意的是，这里的this：

JavaScript 代码:

1. // Lexical this
2. var bob = {
3. \_name: "Bob",
4. \_friends: [],
5. printFriends: function printFriends() {
6. var \_this = this;
8. this.\_friends.forEach(function (f) {
9. return console.log(\_this.\_name + " knows " + f);
10. });
11. }
12. };

从转译结果的printFriends函数中var \_this = this;可以看出，\_this指向bob对象。

比较一下下面两段代码：

JavaScript 代码:

1. var bob = {
2. \_name: "Bob",
3. \_friends: ["愚人码头"],
4. printFriends() {
5. this.\_friends.forEach(f =>
6. console.log(this.\_name + " knows " + f));
7. }
8. };
9. bob.printFriends();// Bob knows 愚人码头

JavaScript 代码:

1. var bob = {
2. \_name: "Bob",
3. \_friends: ["愚人码头"],
4. printFriends() {
5. this.\_friends.forEach(function (f) {
6. return console.log(this.\_name + " knows " + f);
7. });
8. }
9. };
10. bob.printFriends();// undefined knows 愚人码头

原因很简单：箭头函数与其上下文代码共享相同的词法this，所以this指向bob对象，第2段代码forEach中的匿名函数上下文是window对象，所以this指向window对象。

题外话，要想修复第2段代码的问题，除了Babel提供的转译外，还可以使用 ES5 的 bind() ：

JavaScript 代码:

1. var bob = {
2. \_name: "Bob",
3. \_friends: ["愚人码头"],
4. printFriends() {
5. this.\_friends.forEach(function (f) {
6. return console.log(this.\_name + " knows " + f);
7. }.bind(this));
8. }
9. };
10. bob.printFriends();// Bob knows 愚人码头

bind() 最简单的用法是使函数的上下文指向其参数。（<http://www.css88.com/archives/5611>）。