在学习默认参数的时候发现一个bug

**let** foo **=** 'outer';

**function** bar(func **=** x **=>** foo) {

**let** foo **=** 'inner';

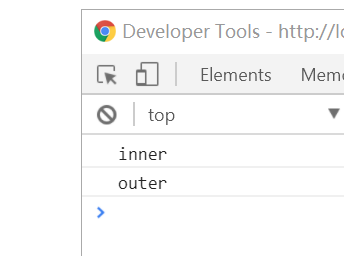
console.log(func()); }

bar(); *//outer*

Babel转义出来

'use strict';  
  
*var* foo = 'outer';  
  
*function bar*() {  
 *var* func = arguments.length > 0 && arguments[0] !== undefined ? arguments[0] : *function* (x) {  
 *return* foo;  
 };  
  
 *var* foo = 'inner';  
 console.*log*(func());  
}  
  
*bar*(); *//inner*

实测



Babel的例子我们很容易理解，有点类似于闭包的概念

Es6的请参考https://www.zhihu.com/question/60394151/answer/176834242

If default value parameter initializers exist, a second [Environment Record](https://link.zhihu.com/?target=http://www.ecma-international.org/ecma-262/7.0/%23sec-environment-records" \t "https://www.zhihu.com/question/60394151/answer/_blank) is created for the body declarations.

A separate Environment Record is needed to ensure that closures created by expressions in the formal parameter list do not have visibility of declarations in the function body.

上面说的意思就是默认初始化是在一个独立的闭包里面初始化而bebal把这个区域放到了内部。