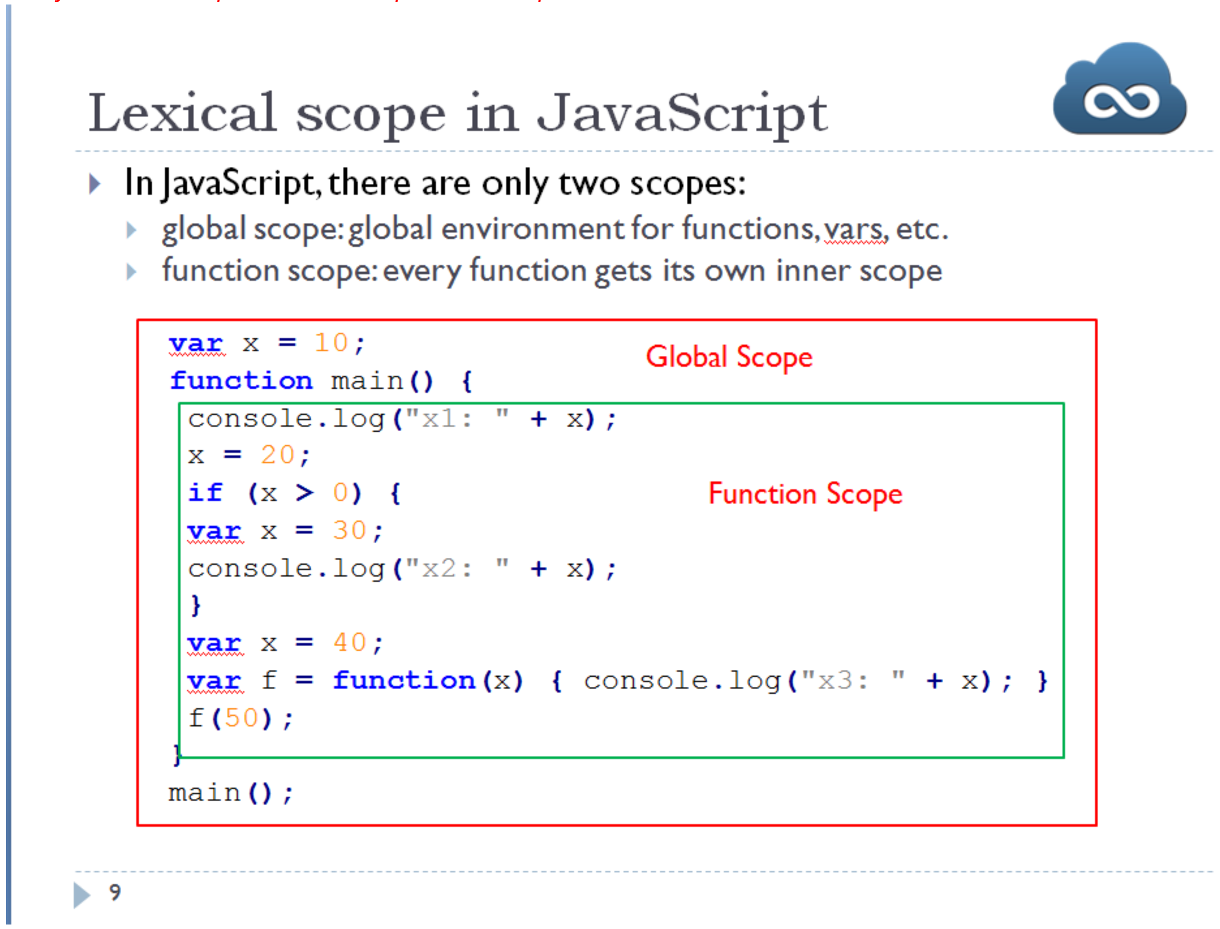
Javascript Module Exercises

1. Determine what this Javascript code will print out (without running it):

```
undefined 8 8 9 10 1
```

https://jsfiddle.net/TinaXing/rct9t3yr/

2. Define Global Scope and Local Scope in Javascript.



Lexical scope in JavaScript

- In JavaScript, there are only two scopes:
  - global scope: global environment for functions, vars, etc.
  - function scope: every function gets its own inner scope

```
var x = 10;
function main() {
 console.log("x1: " + x);
 x = 20;
 if (x > 0) {
 var x = 30;
 console.log("x2: " + x);
 }
 var x = 40;
 var f = function(x) { console.log("x3: " + x); }
 f(50);
}
main();
```

9

In ES6, when use `let` or `const` define a variable, it'll be block scope like Java.

3. Consider the following structure of Javascript code:

```
// Scope A
function XFunc () {
      // Scope B
      function YFunc () {
            // Scope C
      };
};
```

(a) Do statements in Scope A have access to variables defined in Scope B and C? NO
(b) Do statements in Scope B have access to variables defined in Scope A? YES
(c) Do statements in Scope B have access to variables defined in Scope C? NO
(d) Do statements in Scope C have access to variables defined in Scope A? YES
(e) Do statements in Scope C have access to variables defined in Scope B? YES

4. What will be printed by the following (answer without running it)?

```
var x = 9;
function myFunction() {
    return x * x;
}
document.write(myFunction());
x = 5;
document.write(myFunction());
```

Answer: 81 25

5.

```
var foo = 1;
function bar() {
        if (!foo) {
                var foo = 10;
        }
        alert(foo);
}
bar();
```

What will the alert print out? (Answer without running the code. Remember 'hoisting'.)?

Answer: 10
Hoisting foo in bar() execution environment, foo = undefined, undefined is falsey value, so foo gets assigned to 10.

6. Consider the following definition of an add( ) function to increment a counter variable:

```
var add = (function () {
    var counter = 0;
    return function () {
            return counter += 1;
            }
    })();
```

Modify the above module to define a count object with two methods: add( ) and reset( ). The count.add( ) method adds one to the counter (as above). The count.reset( ) method sets the counter to 0.

Answer: https://jsfiddle.net/TinaXing/8uurLbzd/

7. In the definition of add( ) shown in question 6, identify the "free" variable. In the context of a function closure, what is a "free" variable?

Answer: `counter` is free variable.

8. The add( ) function defined in question 6 always adds 1 to the counter each time it is called. Write a definition of a function make_adder(inc), whose return value is an add function with increment value inc (instead of 1). Here is an example of using this function:

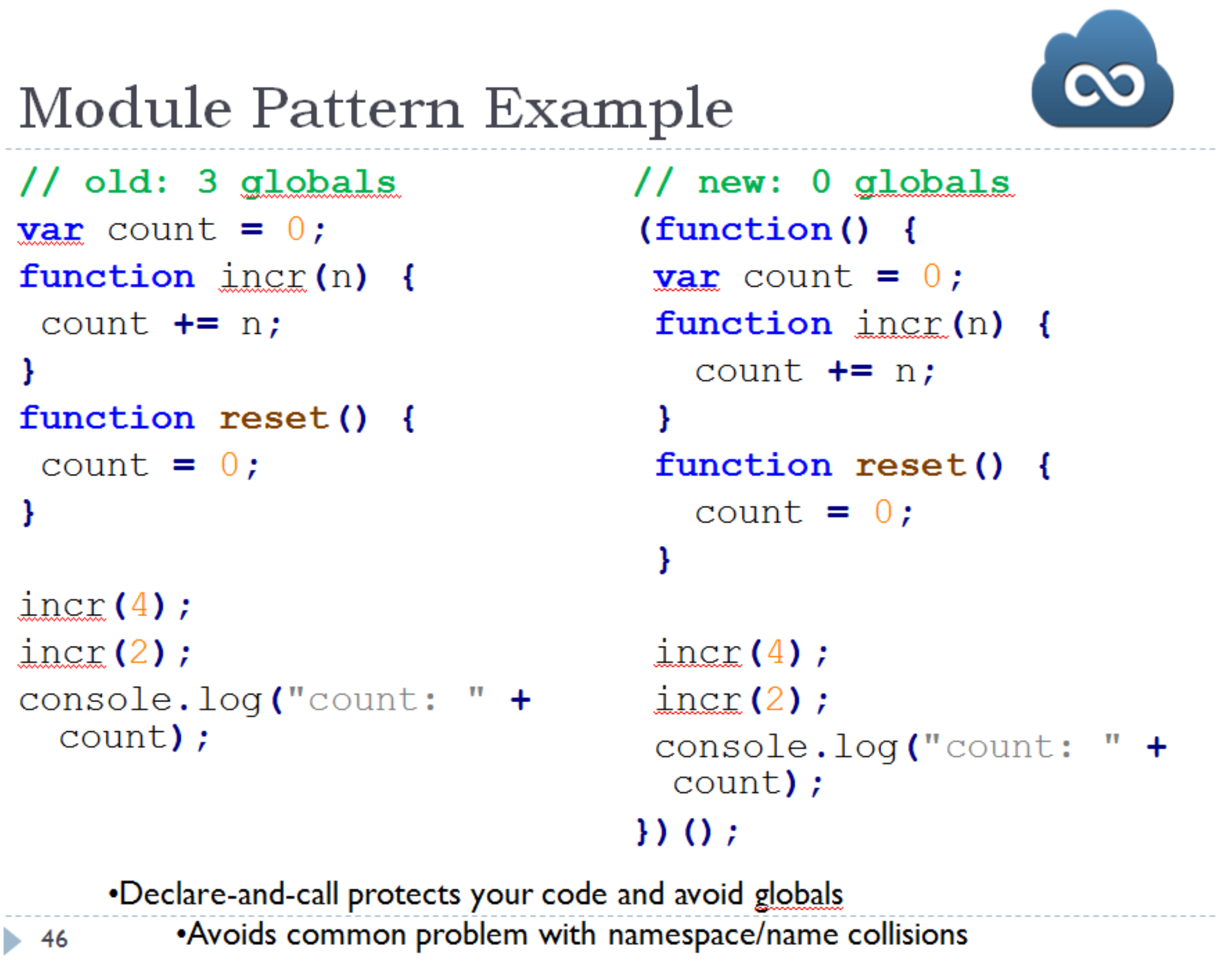
```
add5 = make_adder(5);
add5( ); add5( ); add5( ); // final counter value is 15
add7 = make_adder(7);
add7( ); add7( ); add7( ); // final counter value is 21
```

Answer: https://jsfiddle.net/TinaXing/yqeaesr2/

9. Suppose you are given a file of Javascript code containing a list of many function and variable declarations. All of these function and variable names will be added to the Global Javascript namespace. What simple modification to the Javascript file can remove all the names from the Global namespace?

Answer:

1) Use IIFE



Module Pattern Example

```
// old: 3 globals
var count = 0;
function incr(n) {
 count += n;
}
function reset() {
 count = 0;
}

incr(4);
incr(2);
console.log("count: " +
  count);
```

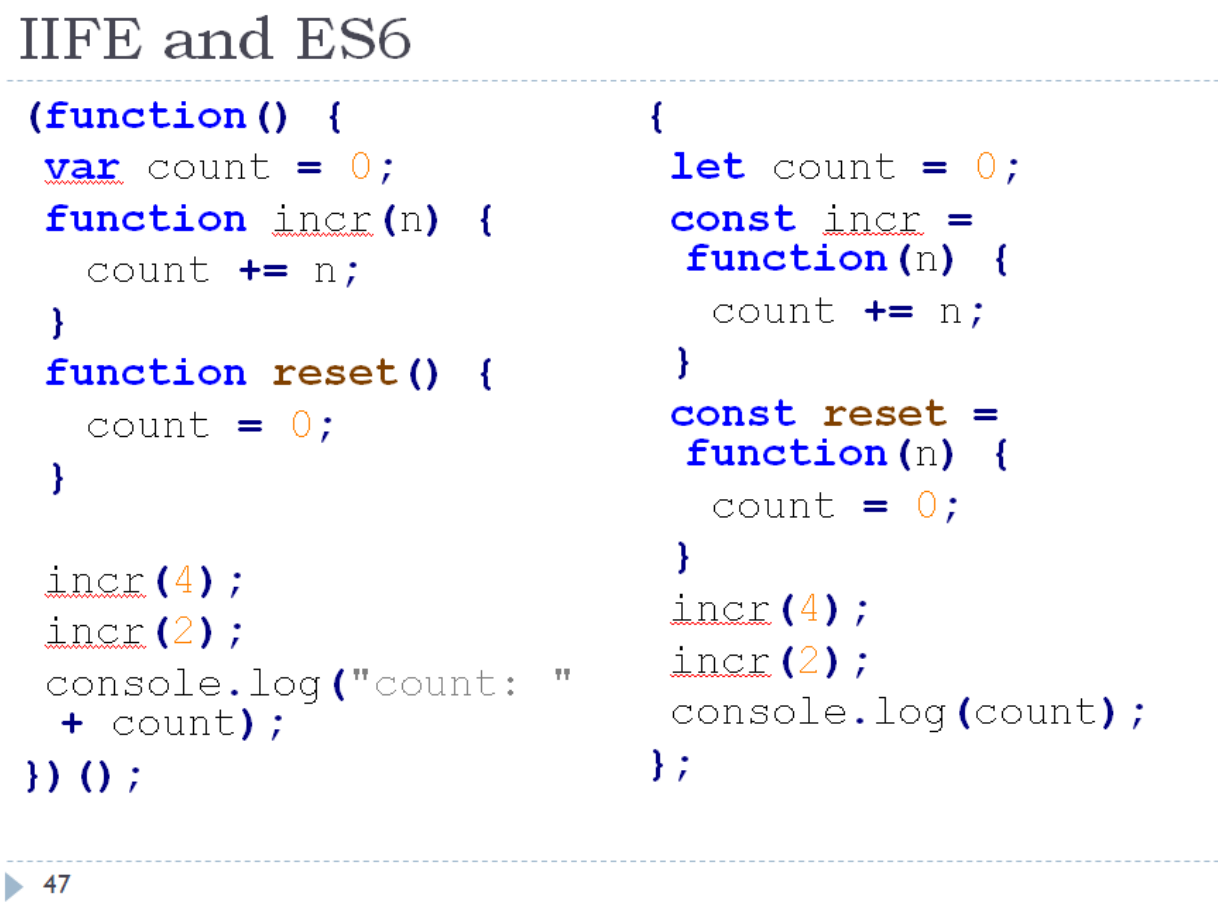
```
// new: 0 globals
(function() {
 var count = 0;
 function incr(n) {
   count += n;
 }
 function reset() {
   count = 0;
 }

 incr(4);
 incr(2);
 console.log("count: " +
  count);
})();
```

•Declare-and-call protects your code and avoid globals

•Avoids common problem with namespace/name collisions

46

2) Use ES6 syntax



IIFE and ES6

```
(function() {
 var count = 0;
 function incr(n) {
   count += n;
 }
 function reset() {
   count = 0;
 }

 incr(4);
 incr(2);
 console.log("count: "
  + count);
})();
```

```
{
 let count = 0;
 const incr =
  function(n) {
   count += n;
 }
 const reset =
  function(n) {
   count = 0;
 }
 incr(4);
 incr(2);
 console.log(count);
};
```

47

10. Using the Revealing Module Pattern, write a Javascript definition of a Module that creates an Employee Object with the following fields and methods:
Private Field: name
Private Field: age
Private Field: salary
Public Method: setAge(newAge)
Public Method: setSalary(newSalary)
Public Method: setName(newName)
Private Method: getAge( )
Private Method: getSalary( )
Private Method: getName( )
Public Method: increaseSalary(percentage) // uses private getSalary( )
Public Method: incrementAge( ) // uses private getAge( )

Answer:
https://jsfiddle.net/TinaXing/0d1wfq3x/

11. Rewrite your answer to Question 10 using the *Anonymous Object Literal Return Pattern*.
Out of Scope

12. Rewrite your answer to Question 10 using the *Stacked Locally Scoped Object Literal Pattern*.
Out of Scope

13. Write a few Javascript instructions to extend the Module of Question 10 to have a public address field and public methods setAddress(newAddress) and getAddress( ).

Answer:
https://jsfiddle.net/TinaXing/0d1wfq3x/