9/27/2020

W2D6 Closure Review Questions

1. Can functions access variables outside the function? Give an example.

1.1 Can code access local variables of an inner function or code block?

2. If the function is assigned as a callback in a separate context does it still have access to the outside variables?

2a. What value do they get if the outside variable is changed before they are called, the most recent value or the value from the time the function was created?

2b. Does it have access to the outer variables in the new context? Explain with reference to slide 7. (“Callbacks with reference to outer variable?”)

3. Explain this statement from slide 11 (“Lexical environment”): To get or change a variable” means “to get or change a property of that object

4. How do lexical environments relate to 'scope' of a variable?

5. What is the lexical environment for 'shooter' when it is created

```
function makeArmy() {
 let shooters = [];
 let i = 0;
 while (i < 10) {
        let shooter = function() { // shooter function
          alert( i ); // should show its number
        };
        shooters.push(shooter);
        i++;
 }
 return shooters;
}
let army = makeArmy();
```

6. What is a nested function?

7. Can nested functions be the value of a return statement? Passed as an argument in a call to another function? Assigned as the value of an object property?

8. Answer the 2 questions on slide “Nested functions 2”.

8.1. How is counter like an object with a private property?

9. The “Environments in detail” slide talks about calling a function and creating its Lexical Environment. Explain what happens in terms of the hidden property [[Environment]] when a function is called.

10. What is the definition of a closure?

11. What is a free variable in a function?

12. What does it mean to expose an inner function?

13. Do JavaScript closures access free variables by value or by reference?

14. What is a code block and what is the scope of any variables declared inside the block?

15. What is a bare code block and why is it useful in JavaScript?

16. What is the global object?

17. How are Lexical Environments related to Execution Contexts and the Execution Context Stack (call stack)?

EC

1. What does it mean to say functions are "action objects"?

2. What are the name and length properties of functions?

3. How are function properties related to closures?