**W3D1 Call Context Review Questions**

1. What is the difference of setTimeout and setInterval
2. What is the difference of setTimeout(sayHi(), 1000); and setTimeout(sayHi, 1000); ?
3. What is returned by setTimeout(sayHi(), 1000);

What is the value of ‘this’ in a method versus in a function? How about in strict mode?

**W2D6 Closure Review Questions**

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

11. What is a free variable in a 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?

**W2D3\_4 Recursion Review Questions**

1. What is the base case in recursion?

2. What is the reduction step?

6. Describe the execution stack for the recursive call pow(2, 3) on slide 8.

14. Describe in words how to

prepend a new item to the beginning of a list;

insert a new item into the middle;

add a new item to the end;

remove an item from the middle;

15. What happens to parameters that have missing arguments?

16. What happens if there are extra arguments?

19. How is the spread operator related to rest parameters?

20. What is the typical usage scenario for rest parameters?

21. What is the typical usage scenario for spread operators?

**W2D2 Review questions for Desturcturing, Date, JSON**

1. Explain what happens on both sides of the = sign for the following destructuring assignment:  
   let [firstName, surname] = "Ilya Kantor".split(' ‘);
2. Explain what happens for the following destructuring assignment, and especially what …remaining means:  
   let [name1, name2, ...remaining] = ["Julius", "Caesar", "Consul", "of the Roman Republic"];
3. Just based on your knowledge of JavaScript language rules, what data type should be returned by parseExpression (below)? What properties should it have? What can we say about expr and rest? Is {expr, rest} an object literal in the following statement?  
   let {expr, rest} = parseExpression(program);
4. What does new Date() return?
5. What is the purpose of JSON.stringify? Can you think of a use case for it?
6. What is the purpose of JSON.parse? Can you think of a use case for it?

**W1D4 Review questions for lesson 4 Data Types**

1. What is the difference of slice and splice?
2. What is a destructive versus non-destructive array operation?
3. How is forEach related to for .. of ?
4. Why are find, filter, map, reduce considered ‘higher order’ functions? Why are they considered “pure” functions?
5. What is the accumulator in reduce?
6. Why do we recommend using an initial value for reduce?

**W1D3 Review questions for lesson 3: Objects**

1. What is a method? How is it related to a function?
2. What is ‘this’ in an object? When is it used?
3. What is the value of this if called in a function that is not a method?
4. What is a constructor function? How does it relate to an object literal?
5. What does the operator ‘new’ do when called with a constructor function?
6. What happens if you forget to use ‘new’ when calling a constructor function?
7. Why are constructor functions capitalized? What happens if they are not capitalized?

W2D5 Review Assignment

Try to do at least the first two, and then the others as time allows.

1. Write your own version of map. Write a function, myMap that takes 2 arguments, an array and a function to apply to the array. It should return a new array of the same size with the function applied to each element of the input array. It should not change the input array.
2. Write your own version of filter. Write a function, myFilter that takes 2 arguments, an array and a function to apply to the array. It should return a new array with the function applied to each element of the input array. It should not change the input array. It should work like Array.filter. I.e., the input function returns true or false for each element in the original array, and the true elements are included in the returned array.
3. Write your own version of reduce.
4. Write a constructor function to replace the Class given for the Simpsons tree recursion exercise.
5. Write a constructor function to create nodes for the HTML DOM tree recursion exercise from the W1D3 recursion assignment.
6. EC: Write your Mocha-like framework. You will need to implement your own Describe and It functions, and an assert-equals function. Do not worry about all the other assert functions that come with Chai. Your framework should work the same way as Mocha in terms of writing outputs to a target div on a webpage. Color the text red for any tests that fail and green for those that succeed.

* Review any questions you missed on the first exam. They might appear again
* Write own version of map/filter/reduce sort of Array methods
* Use Array.reduce to do things like find min/max of numbers or objects
* Simple recursion, e.g., factorial or the tree walk (log to console) or treeModifier or findNode exercises (5 and 6 of W2D4)
  + write recursive code to add a property to all nodes in a tree structure (or log some value of all nodes) ..
* Linked list manipulation exercises from W2D4 (e.g., insert node or remove node)
* Simple usage of spread or rest or destructure
* Filter through function exercise (we coded one in class)
* Simple closure code such as makeCounter
* Timer exercise such as the stopWatch example