9/27/2020

W2D3_4 Recursion Review Questions

1. What is the base case in recursion?

2. What is the reduction step?

3. What is the execution context of a running function?

4. What is the execution (runtime) stack? What is in it and how does it operate?

5. Describe what will be on the execution stack when this line of code runs: [1, 2].map(item => item + 10);

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

7. What are the usual use cases for using recursion?

8. What is a recursive data structure?

9. Why are HTML documents recursive data structures?

10. Why are linked lists recursive data structures?

11. Why is a linked list better than an array for a large queue?

12. Why are linked lists better than arrays for use cases that have frequent insertions and deletions?

13. Describe in words how to split and rejoin a linked list.

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?

17. What is 'overloading' of a function and why does it not normally happen in JavaScript?

18. How do rest parameters relate to the arguments object? Which one is an array?

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?

22. How do spread operators relate to object destructuring?