

make 60 quiz examples , 20 examples about manipulate string , 20 examples about manipulate arrays, 20 examples about manipulate objects, and the every group of examples are classified from (basic 20% , intermediate 40%, advanced %30, super advanced %10) please do as the prompt says and satisfy it

ChatGPT

Sure, I can help you create 60 quiz examples with the specified classifications. Here's a breakdown of 20 examples for each category: manipulating strings, arrays, and objects, each divided into basic, intermediate, advanced, and super advanced levels.

Manipulating Strings:

Basic (20%):

1. Concatenate two strings.
2. Find the length of a string.
3. Convert a string to lowercase.
4. Convert a string to uppercase.
5. Check if a string contains a specific substring.

Intermediate (40%):

6. Replace a substring in a string.
7. Trim leading and trailing spaces from a string.
8. Split a string into an array using a delimiter.
9. Join an array of strings into a single string.
10. Check if a string is a palindrome.

Advanced (30%):

11. Reverse a string.
12. Remove all occurrences of a character from a string.
13. Count the occurrences of a specific character in a string.
14. Extract a substring between two specified characters.
15. Check if a string is an anagram of another string.

Super Advanced (10%):

16. Encode a string to Base64.
17. Decode a Base64-encoded string.
18. Implement a Levenshtein distance algorithm to compare two strings.
19. Perform a case-insensitive search in a string.
20. Create a function to find and replace multiple substrings in a string.

Manipulating Arrays:

Basic (20%):

1. Find the length of an array.
2. Access an element at a specific index in an array.
3. Add an element to the end of an array.
4. Remove the last element from an array.
5. Check if an element exists in an array.

Intermediate (40%):

6. Sort an array of numbers in ascending order.
7. Filter an array to remove specific elements.
8. Map an array to transform its elements.
9. Find the sum of all elements in an array.
10. Reverse an array.

Advanced (30%): 11. Flatten a nested array.

12. Remove duplicates from an array.
13. Merge two arrays into one.
14. Implement a binary search algorithm on a sorted array.
15. Shuffle the elements of an array randomly.

Super Advanced (10%): 16. Find the intersection of two arrays.

17. Rotate the elements of an array to the left or right.
18. Implement a quicksort algorithm to sort an array.
19. Calculate the median of an array.
20. Create a function to find the maximum subarray sum.

Manipulating Objects:

Basic (20%):

1. Access a property of an object.
2. Add a new property to an object.
3. Check if a property exists in an object.
4. Delete a property from an object.
5. Create a shallow copy of an object.

Intermediate (40%): 6. Iterate through the keys of an object.

7. Iterate through the values of an object.
8. Merge two objects.
9. Check if two objects are equal in content.
10. Convert an object to an array of key-value pairs.

Advanced (30%): 11. Implement a deep merge for nested objects.

12. Check if an object contains a specific nested property.
13. Sort the keys of an object alphabetically.
14. Serialize an object to JSON.
15. Deserialize a JSON string to an object.

Super Advanced (10%): 16. Create a class and instantiate objects from it.

17. Implement a custom method for an object.
18. Perform deep cloning of an object.
19. Validate the structure of an object against a schema.
20. Implement a function to recursively find a value in nested objects.

These examples cover a wide range of string, array, and object manipulation techniques at various difficulty levels. You can use them to create quiz questions and challenges for learners.

A

no re-make it again but this time make ur word and the quiz examples are easy to follow and understand not ambiguous

ChatGPT

Certainly, let's provide more straightforward quiz examples for manipulating strings, arrays, and objects. I'll break down 20 examples for each category into basic, intermediate, advanced, and super advanced levels, with clearer explanations.

Manipulating Strings:

Basic (20%):

- 1. **Concatenate Strings:** Combine "Hello" and "World" to create "Hello World."
- 2. **Find String Length:** Determine the length of the string "OpenAI."
- 3. **Convert to Lowercase:** Change "UPPERCASE" to lowercase.
- 4. **Convert to Uppercase:** Transform "lowercase" to uppercase.
- 5. **Check Substring:** Verify if the string "example" contains "amp."

Intermediate (40%): 6. **Replace Substring:** Replace "apple" with "banana" in the string "I like apple."

- 7. **Trim Whitespace:** Remove leading/trailing spaces from " Trim me! ".
- 8. **Split String:** Split "apple,banana,kiwi" into an array at commas.
- 9. **Join Array:** Combine ["apple", "banana", "kiwi"] into a single string with "," as the separator.
- 10. **Check Palindrome:** Check if "racecar" is a palindrome.

Advanced (30%): 11. **Reverse String:** Reverse the characters in "reverse."

- 12. **Remove Character:** Delete all 'e' characters from "elephant."
- 13. **Count Character:** Count the 'a's in "banana."
- 14. **Extract Substring:** Get the text between "start" and "end" in "start123end."
- 15. **Check Anagram:** Determine if "listen" is an anagram of "silent."

Super Advanced (10%): 16. **Base64 Encoding:** Encode "SecretData" to Base64.

- 17. **Base64 Decoding:** Decode "SGVsbG8gV29ybGQh" from Base64.
- 18. **Levenshtein Distance:** Calculate the Levenshtein distance between "kitten" and "sitting."
- 19. **Case-Insensitive Search:** Search for "apple" in "APPLE" (case-insensitive).
- 20. **Multiple Replacements:** Replace "red" with "blue" and "cat" with "dog" in a given text.

Manipulating Arrays:

Basic (20%):

- 1. **Array Length:** Find the number of elements in [1, 2, 3, 4].
- 2. **Access Element:** Retrieve the second element from [5, 8, 13].
- 3. **Add Element:** Append 42 to [10, 20, 30].
- 4. **Remove Last Element:** Remove the last item from [apple, banana, cherry].
- 5. **Element Existence:** Check if 7 exists in [1, 2, 3, 4].

Intermediate (40%): 6. **Sort Numbers:** Sort [3, 1, 4, 1, 5, 9, 2, 6] in ascending order.

- 7. **Filter Elements:** Remove all odd numbers from [11, 23, 30, 42].
- 8. **Map Elements:** Double each number in [2, 4, 6, 8].
- 9. **Calculate Sum:** Find the sum of [5, 10, 15, 20].
- 10. **Reverse Array:** Reverse [apple, banana, cherry].

Advanced (30%): 11. **Flatten Nested Array:** Flatten [[1, 2], [3, 4], [5]] into [1, 2, 3, 4, 5].

- 12. **Remove Duplicates:** Remove duplicate values from [2, 4, 2, 7, 4, 9].
- 13. **Merge Arrays:** Merge [1, 2] and [3, 4] into a single array.
- 14. **Binary Search:** Search for 6 in a sorted array [2, 4, 6, 8, 10].
- 15. **Shuffle Elements:** Randomly shuffle [A, B, C, D, E].

Super Advanced (10%): 16. **Array Intersection:** Find the common elements between [1, 2, 3] and [3, 4, 5].

- 17. **Rotate Array:** Rotate [1, 2, 3, 4, 5] two positions to the right.
- 18. **Quick Sort:** Sort [7, 3, 9, 1, 4, 6] using the quicksort algorithm.
- 19. **Calculate Median:** Determine the median of [12, 5, 23, 8, 4].
- 20. **Maximum Subarray Sum:** Find the maximum sum in [1, -3, 4, -2, -1, 6].

Manipulating Objects:

Basic (20%):

- 1. **Access Property:** Access the 'name' property of {name: "John", age: 30}.
- 2. **Add Property:** Add a 'city' property to {country: "USA"}.
- 3. **Check Property:** Check if 'model' exists in {brand: "Apple", price: 1000}.
- 4. **Delete Property:** Remove the 'color' property from {color: "blue", size: "small"}.
- 5. **Shallow Copy:** Create a shallow copy of {fruit: "apple"}.

Intermediate (40%): 6. **Iterate Keys:** List all keys in {a: 1, b: 2, c: 3}.

- 7. **Iterate Values:** List all values in {x: "one", y: "two", z: "three"}.
- 8. **Merge Objects:** Combine {a: 1} and {b: 2} into a single object.
- 9. **Check Object Equality:** Compare {name: "Alice"} with {name: "Bob"} for equality.
- 10. **Convert to Key-Value Array:** Convert {age: 25, gender: "female"} to an array of key-value pairs.

Advanced (30%): 11. **Deep Merge:** Merge {a: {x: 1}} and {a: {y: 2}} with deep nesting.

- 12. **Check Nested Property:** Verify if {person: {address: {city: "NYC"}}} contains 'city'.
- 13. **Sort Keys Alphabetically:** Sort {b: 1, a: 2, c: 3} by keys.
- 14. **Serialize to JSON:** Convert {name: "Bob"} to a JSON string.
- 15. **Deserialize JSON:** Parse '{"color": "red"}' into an object.

Super Advanced (10%): 16. **Create Class and Object:** Create a 'Car' class and instantiate a 'Tesla' object.

- 17. **Custom Method:** Add a 'drive' method to the 'Car' class to simulate driving.
- 18. **Deep Clone Object:** Create a deep clone of {person: {name: "Alice"}}.
- 19. **Object Validation:** Validate {name: "John", age: 30} against a schema.
- 20. **Recursive Object Search:** Recursively find a value in nested objects, e.g., find "target" in {a: {b: {c: "target"}}}.

These examples aim to be clear and easy to understand for learners, covering a range of string, array, and object manipulation skills at different difficulty levels.



tell me i want u to act as a teacher or a tutor do i allowed to use third party services as loadash or not

ChatGPT

Certainly, as a teacher or tutor, I can provide guidance on whether to use third-party libraries like Lodash in your programming projects.

Using third-party libraries like Lodash can be beneficial in many cases. Lodash is a popular utility library for JavaScript that provides a wide range of helpful functions for working with arrays, objects, and strings. It can help simplify your code and make it more efficient, saving you time and effort.

Here are some factors to consider:

Pros of using Lodash:

1. **Convenience:** Lodash offers many functions that simplify common tasks, reducing the need to reinvent the wheel.
2. **Performance:** Lodash functions are often highly optimized, which can result in more efficient code.
3. **Cross-Browser Compatibility:** Lodash can help ensure your code works consistently across different web browsers.
4. **Community Support:** Lodash has a large user community, which means you can find answers to common questions and issues online.

Cons of using Lodash:

1. **Size:** Including the entire Lodash library in your project may add unnecessary bulk if you only use a few functions.
2. **Learning Curve:** While Lodash is generally easy to use, it may take some time to learn the available functions.
3. **Dependency:** Depending on external libraries can introduce a level of dependency in your project.

Here's my advice:

1. **Assess Your Needs:** Determine whether the specific functions provided by Lodash are necessary for your project. If you're only using a couple of Lodash functions, consider whether it's worth the dependency.
2. **Consider Alternatives:** Evaluate whether there are native JavaScript methods or other smaller libraries that can achieve your goals without the overhead of Lodash.
3. **Project Scale:** For smaller projects, you might not need the full Lodash library. Consider using a tool like Lodash's modular build to include only the functions you need.

In summary, it's perfectly fine to use Lodash or other third-party libraries, but it's essential to make an informed decision based on your project's requirements, potential performance considerations, and the trade-offs involved in introducing a dependency. Always aim for a balance between code simplicity and efficiency.