Saving... Preview

Memoization

Pre-Class Check List for Instructor

Pre-Class Checklist [For Instructor]

👉Please keep the camera on, and ensure it's on during the whole lecture.
👉Please go through the In-class content before the lecture (Lecture Notes, In class assessments

Guidelines

👉Instructions are enclosed within the square brackets[].
👉Relevant material is marked in bold.
👉We humbly expect the instructor to deliver the topic in the suggested delivery style of the content following the instruction.
👉The code along activities included need to be followed step by step.
👉Make sure to start the class full of enthusiasm and switch on the camera for the best interaction.

Estimated Timeline/Topic

Saving...

ON THIS PAGE

Pick up an expensive function and explain it - 10 min

Now introduce memoization - 10 min

Code along the memoized function - 30min

Dry run using a test case - 20min

Use cases - 20min

Summary Quiz - 10min

Lecture Flow

Welcome Note

[Welcome Note]

👉 "Hi, I am 'XYZ', an instructor at Newton School. I am so happy to start this journey together with you".

Motivating Students

[Telling students that they need to put real effort to be a good developer and we will be helping them]

👉Learning Coding is similar to the Workout you do. You should invest time consistently to see the changeover.

👉Make sure you learn something new every day even if it's something small.

Saving...

to becoming a Good Developer"

ON THIS PAGE

Lecture Flow

Start with opening up the VS code and writing an expensive function.
(An expensive function could be any function that takes a good amount of time to execute)

For eg.

javascript

Write your code below

Write your code in the respective coding windows to avoid errors etc etc.
Use SHIFT+ENTER to create new line in code editor.

Pre-function Code

```
1
```

Function Code

```
1  function expensiveOperation(n) {
2    console.log('Performing expensive operation...');
3    return n * 2;
4  }
5
6  console.log(expensiveOperation(10));  //Let's say it takes 2 sec
```

Saving...

Post-function code

1

ON THIS PAGE

Show Post-function and Pre-function code to the reader
The whole code will be shown to the user as a single code block.

User can edit Function Code
The function code will be editable in read-only mode.

Code is Runnable
The user can run the code and see the outcome.

Code Snippet
The code will be shown to user as a read-only snippet

Contd.

Explain them that the expensive function will take the same time even for the same input, as their is no caching mechanism that caches the result and utilize it when the fucntion is called for the same input.

Now introduce them with the word memoization.
Explain its definition :

Saving...

ON THIS PAGE

expensive function calls and returning the cached result when the same inputs occur again. It is a form of caching that helps improve performance by avoiding redundant computations.

So using this point explain the students the importance of memoizing the function for some input value. Explain them how it will enhance the performance of their code.

Now show them the same by writing the memoized code.

javascript

Write your code below

Write your code in the respective coding windows to avoid errors etc etc.
Use SHIFT+ENTER to create new line in code editor.

Pre-function Code

```
1
```

Function Code

```
function memoize(func) {
  const cache = {};

  return function(...args) {
    const key = JSON.stringify(args);

    if (cache[key]) {
      return cache[key];
    }

    const result = func.apply(this, args);
    cache[key] = result;

```

Saving...

```
// Example function to be memoized
function expensiveOperation(n) {
  console.log('Performing expensive operation...');
  return n * 2;
}

const memoizedOperation = memoize(expensiveOperation);

console.log(memoizedOperation(5)); // Output: Performing expensive operation... 10
console.log(memoizedOperation(5)); // Output: 10 (Result retrieved from cache)
```

ON THIS PAGE

Post-function Code

```
```

Show Post-function and Pre-function code to the reader

The whole code will be shown to the user as a single code block.

User can edit Function Code

The function code will be editable in read-only mode.

Code is Runnable

The user can run the code and see the outcome.

Code Snippet

The code will be shown to user as a read-only snippet

Contd.

Saving...

ON THIS PAGE

Now make them realize how exactly this function is memoizing the input value by caching the result for every input to avoid redundant call for the same input value.

Explain the above code something like this :

In the above example, the memoize function takes a function (expensiveOperation) as an argument and returns a memoized version of that function. The memoized function uses an internal cache object to store the results of previous function calls.
When the memoized function is called with certain arguments, it first checks if the result for those arguments exists in the cache. If it does, the cached result is returned directly. Otherwise, the original function is called with the arguments, and the result is stored in the cache before being returned.

In subsequent calls to the memoized function with the same arguments, the cached result is retrieved, avoiding the expensive computation and improving the performance.

Memoization can be beneficial when dealing with functions that have expensive calculations or operations that are repeated with the same inputs. It can significantly reduce the execution time and improve the efficiency of your JavaScript programs.

Ending of the lecture

👉 Start the summary quiz and ask students to attempt the same.

Saving...

performance and optime the time complexity of your code.

ON THIS PAGE