JS NUGGETS

MEMOIZATION IN JAVASCRIPT

IDEAS2IT

Memoization is an optimization technique used to speed up programs by **storing the results of function calls** and returning the cached result when the same inputs occur again.

It works if the result of calling a function with the same set of arguments results in the same output. In other words, the function must be pure, otherwise caching the result would not make sense.

Why is this useful? 🤔

Suppose our function takes 1 second to run, while caching lets us speed up the process to 2 milliseconds. 

IDEAS2IT

When to memoize function? 

- The best use case of memoized functions is **for heavy computational functions.**
- Function should be pure so that it returns the same output each time they are called with a particular input.
- Functions should have a limited input range so that cached values can be made use of more frequently.
- Recursive functions with recurring input values.

IDEAS2IT

Example😀

```
const memoizedAddition = () => {
 let cache = {};
 return (n) => {
   if (n in cache) {
     console.log('Fetching from cache');
     return cache[n];
   }
   else {
     console.log('Calculating result');
     const result = n + 1;
     cache[n] = result;
     return result;
   }
 }
}
const addition = memoizedAddition();
console.log(addition(7)); // Calculating result 8
console.log(addition(7)); // Fetching from cache 8
console.log(addition(7)); // Fetching from cache 8
```