

## **JAVASCRIPT CHEAT SHEET FOR INTERVIEWS & TECH ROUNDS**

Javascript a server-side scripting language or client-side scripting language?

Why is javascript a weakly typed language?

Storage structure of primitive data types & non primitive data types in js?

Explain by code example why js is phrased out as "Everything is an object in js"?

What are prototypes? Describe prototype functions?

What is prototype chaining?

What is a block ? What is scope?

What do you mean by the term global context?

What is a call stack?

What changes happen in a call stack when a function call is made?

What is the lexical environment ?

Explain the term Currying?

What is a Callback queue? Microtask queue in js?

Window ?

BOM vs DOM?

What is scope chaining ?

Shadowing? Show with example ILLEGAL SHADOWING ?

Difference between let vs const vs var ?

Type coercion in js?

Explain =>

1. For of loop
2. For in loop
3. Foreach loop

Differentiate between '==' and '==='?

Types of functions in js ?

What do you mean by the term Anonymous function?

What do you mean by the term Higher Order Functions and first class functions?

Differentiate between function statement, function expression & function declaration?

How does ' this' keyword behave when creating arrow functions and normal functions?

Explain the concept of map, filter and reduce with examples?

Hoisting in js?

Minimal valid code structure in js ?

Closures ? How can an external variable that is capable of accessing members of a closure function be explained with call stack chaining?

setTimeout() vs setInterval()?

setTimeout(()=>{},0) vs ()=>{}?

setTimeout(()=>{},1000) vs setTimeout(()=>{},0)

When does setTimeout behave wrongly? Why does that happen?

Event loop ?

Explain with scenarios

1 when this is equivalent to window

2. When this is not equivalent to window

Event listeners?

Event Listeners vs Event handlers?

Why are event listeners termed as heavy tasks? And need to be canceled after it is used?

Event capturing? Event trickling?

Event bubbling?

Event delegations? why should we use them?

Explain callbacks in js?

Pyramid of Doom problem?

Inversion of control?

Callback hell?

Explain the problem caused by callback chaining and how the promises concept solves that problem?

What are promises ?

Promises vs callbacks?

Exception handling in promises?

Derive a promise function making an api call after a time delay of 2 seconds ?

Async & await ?

## **JS RUNTIME ENVIRONMENT SPECIFIC TOPICS**

Js engine ? What are the parts of js engine?

Can js run on any other platforms other than browsers?

Js engine is software or hardware?

What are web apis?

Window ?

Callback queue?

Microtask queue?

Event loop?

Browser?

## **ADVANCE TOPICS**

What do you mean by the term temporal dead zone?

Why is strict mode used?

What are the different ways a function can be called in js?

When to use constructor function calling rather than normal function calling?

Explain the difference between generators & iterators?

What is the difference between normal inheritance and prototype inheritance?

Call vs apply vs bind

Derive your own

1. (Call ) custom function?
2. (Apply) custom function?
3. (Bind) custom function?

What are decorators in js? Why are they used?

How is memorization related to decorators' concept?

Throttle ? explain in detail?

How to handle multiple clicks events generated upon dom elements ?

How to handle multiple users trying to access DOM at same time?

How to solve race situations during application building?

What is debouncing?

Throttle vs debounce?

How does debouncing concepts help us to synchronize API calls?

Event propagation ?What is the default technique used by browsers?

Spread operators?

Null vs undefined vs not-initialized variables?

How does cloning work in js?

object.freeze() vs const ?

What are IIFE'S?

**@sHADOWCODGEN**

.....featured by Jyotirmai Tiwari :)

