Basic JS every **React** developer should know

SWIPE TO LEARN

RAJ BHENSDADIYA

- Hey there, fellow JavaScript developers!
- Are you ready to take the plunge into the exciting world of React?
- Let's ensure you're armed with the JavaScript fundamentals to make your React journey a breeze.

RAJ BHENSDADIYA

Basics Concepts

RAJ BHENSDADIYA

Scoping

- Understanding scoping is vital. Variables declared inside a function are scoped to that function

```
function myFunction() {
  var localVar = "I'm local"
  console.log(localVar)
}
myFunction()
console.log(localVar) // Error - localVar is not defined
```

Callbacks

- Callbacks are functions passed as arguments to other functions, often used for async tasks

```
function fetchData(callback) {
  // Simulating async fetch
  setTimeout(() => {
    const data = 'Fetched data'
    callback(data)
  }, 1000)
}

fetchData((result) => {
  console.log(result)
})
```

Passing Functions

- Functions can be passed as arguments too

```
function greet(name) {
  console.log(`Hello, ${name}!`)
}

function processUser(greetingFunction) {
  greetingFunction('Alice')
}

processUser(greet) // Output: Hello, Alice!
```

Reference vs Value

- Primitives (value types) are copied by value, objects (reference types) by reference

```
let a = 5
let b = a
a = 10
console.log(b) // Output: 5

let obj1 = { value: 10 }
let obj2 = obj1
obj1.value = 20
console.log(obj2.value) // Output: 20
```

== vs ===

- Remember the difference between loose equality (==) and strict equality (===)

```
console.log(5 == '5') // Output: true
console.log(5 === '5') // Output: false
```

Advanced Concepts

RAJ BHENSDADIYA

Short Circuiting

- Harness short-circuit evaluation to streamline your code

```
const user = {
  name: 'Alice',
  isAdmin: true,
}

const displayName = user.name || 'Guest'
console.log(displayName) // Output: Alice
```

Advanced Array Methods

- Elevate your skills with powerful array methods like map, filter, and reduce

```
const products = [
  { name: 'Widget A', price: 10, quantity: 50 },
  { name: 'Widget B', price: 15, quantity: 30 },
  { name: 'Widget C', price: 8, quantity: 70 },
  // ... more products
]

// Using map to calculate revenue for each product
const productRevenues = products.map(
  (product) => product.price * product.quantity
)

// Using filter to select products with revenue over $500
const highRevenueProducts = products.filter((product) =>
productRevenues > 500)

// Using reduce to calculate the total revenue
const totalRevenue = productRevenues.reduce((acc, revenue) =>
acc + revenue, 0)

console.log('Product Revenues:', productRevenues) // [500, 450,
560]
console.log('High Revenue Products:', highRevenueProducts)
console.log('Total Revenue:', totalRevenue) // 1510
```

Immutability

- Immutable data helps prevent unintended side effects. Use techniques like the spread operator to achieve this

```
const originalArray = [1, 2, 3]
const newArray = [...originalArray, 4]
console.log(newArray) // Output: [1, 2, 3, 4]
```

Asynchronous

- Async operations are at the heart of modern web development. Master promises and async/await

```
function fetchData() {
  return new Promise((resolve) => {
    setTimeout(() => resolve('Data fetched!'), 1000)
  })
}

async function getData() {
  try {
    const result = await fetchData()
    console.log(result) // Output: Data fetched!
  } catch (error) {
    console.error(error)
  }
}
```

RAJ BHENSDADIYA

Concepts of Modules

- Modularize your code for better organization and maintainability

```
// math.js
export function add(a, b) {
  return a + b
}

// app.js
import { add } from './math'
console.log(add(2, 3)) // Output: 5
```

Spread Operator

- ES6 introduced many powerful features. The spread operator is a game-changer

```
const arr1 = [1, 2, 3]
const arr2 = [4, 5, 6]
const mergedArray = [...arr1, ...arr2]
console.log(mergedArray) // Output: [1, 2, 3, 4, 5, 6]
```

I'm Raj.

Software Engineer;
passionate for building and shipping scalable software.

Let's connect and expand our professional network together.

I'm excited to collaborate, exchange ideas, and contribute to impactful projects.

Feel free to reach out and let's create something amazing together.