

# **Complete JavaScript (ES6+) Guide for React Jobs**

# Complete JavaScript (ES6+) Guide for React Jobs

## 1. Arrow Functions

Arrow functions provide a concise syntax for writing functions.

Syntax:

```
const add = (a, b) => a + b;
```

Differences from traditional functions:

- Do not have their own 'this'.
- Cannot be used as constructors.
- Best for callbacks and small functions.

Practice:

1. Convert normal functions to arrow functions.
2. Use arrow functions in array methods like map/filter.

Interview Questions:

- What's the difference between arrow and regular functions?
- When would you avoid using arrow functions?

# Complete JavaScript (ES6+) Guide for React Jobs

## 2. Destructuring

Destructuring allows unpacking values from arrays or objects.

Syntax:

```
const [a, b] = [1, 2];
```

```
const {name, age} = {name: 'John', age: 30};
```

Usage:

- With function arguments.
- When processing API data.

Interview Questions:

- How do you destructure nested objects?
- What happens when the variable is undefined?

# Complete JavaScript (ES6+) Guide for React Jobs

## 3. Spread / Rest Operators

Spread (...) is used to copy and combine arrays/objects.

Rest (...) collects remaining elements into an array.

Examples:

```
const newArr = [...oldArr, 4];
```

```
const {name, ...rest} = user;
```

Interview Questions:

- Difference between spread and rest?
- Can spread be used for deep cloning?

# Complete JavaScript (ES6+) Guide for React Jobs

## 4. Promises & Async/Await

Promises help handle asynchronous operations.

Syntax:

```
const promise = new Promise((resolve, reject) => {...});
```

Async/Await simplifies promise chaining.

```
async function fetchData() {  
  try {  
    const res = await fetch(url);  
  } catch (err) {  
    console.error(err);  
  }  
}
```

Interview Questions:

- How do you handle errors with async/await?
- Difference between then/catch and async/await?

# Complete JavaScript (ES6+) Guide for React Jobs

## 5. Modules (import/export)

JavaScript modules allow code reusability.

```
export const name = 'JS';
```

```
import { name } from './file.js';
```

Types:

- Named exports
- Default exports

Interview Questions:

- How are named and default exports different?
- What happens on circular dependencies?

# Complete JavaScript (ES6+) Guide for React Jobs

## 6. Array Methods

Array methods allow clean data manipulation.

Examples:

- map: transform each element
- filter: return matching elements
- reduce: return single accumulated value

Interview Questions:

- map vs forEach?
- How to remove duplicates from an array using reduce?

# Complete JavaScript (ES6+) Guide for React Jobs

## Practice Projects

1. Todo App - with map, filter
2. API Fetcher - using async/await
3. User Cards - with destructuring and modules
4. Calculator - with module exports
5. Blog Post Viewer - with array methods and Promises

Tip: Use DevTools, console.log, and break down problems into functions.