

JavaScript Interview Cheat Sheet for React Developers

1. Null vs Undefined

- **undefined**: Variable declared but not assigned a value. Default state.
- **null**: Explicit assignment of 'no value'.

Example:

```
let a; // undefined
let b = null; // null
```

2. Hoisting

- Variable and function declarations are moved to top of scope.
- **var**: hoisted & initialized as undefined.
- **let/const**: hoisted but stay in Temporal Dead Zone (TDZ).
- Function declarations fully hoisted.

Example:

```
console.log(a); // undefined
var a = 10;
console.log(b); // ReferenceError
let b = 20;
```

3. Closures

- A closure is when a function 'remembers' variables from its outer scope.
- Used in React hooks (e.g., useState, useEffect).

Example:

```
function outer(){
  let count=0;
  return function(){count++; return count;}
}
const inc = outer();
inc(); // 1
inc(); // 2
```

4. Event Loop

- JS is single-threaded, async handled via event loop.
- Macrotasks: setTimeout, setInterval.
- Microtasks: Promise.then, async/await.

Example:

```
console.log('A');
setTimeout(()=>console.log('B'),0);
Promise.resolve().then(()=>console.log('C'));
console.log('D');
// Output: A D C B
```

5. Immutability in React

- State should not be mutated directly, always return a new copy.

Example:

```
const [arr, setArr] = useState([1,2,3]);
// Wrong: arr.push(4)
```

```
// Correct: setArr([...arr,4])
```

6. Shallow vs Deep Copy

- React re-renders based on shallow comparison.
- Shallow copy: `Object.assign`, `spread (...)`.
- Deep copy: `JSON.parse(JSON.stringify(obj))` or libraries.

7. Equality

- `==` allows type coercion.
- `===` strict equality (no coercion).

Example:

```
null == undefined // true  
null === undefined // false
```

This cheat sheet covers the core JavaScript concepts most often asked in React mid/senior interviews.