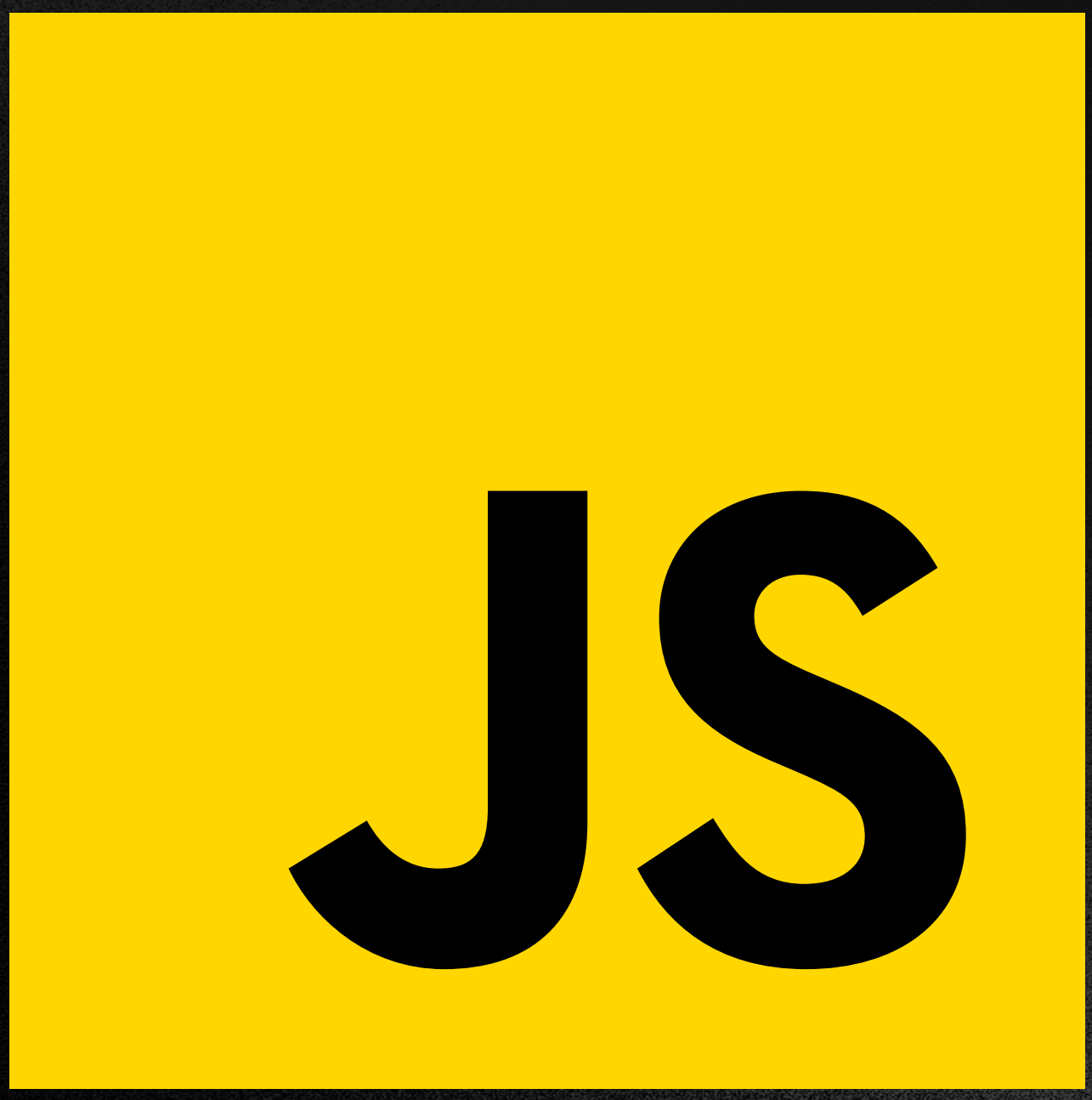
Java Script

“ spread “ Operator

Where to use it ?



[Don’t Miss It]

Cloning Arrays

- Use the spread operator (...) to clone arrays effortlessly.

- Perfect for creating a copy without modifying the original.

JS

```
const originalArray = [1, 2, 3];
const clonedArray = [...originalArray];

console.log(clonedArray);  // Outputs: [1, 2, 3]
```

Concatenating Arrays

- Easily concatenate arrays with the spread operator.
- Say goodbye to concat method verbosity. No more concat method.

JS

```
const array1 = [1, 2, 3];
const array2 = [4, 5, 6];

const combinedArray = [...array1, ...array2];

console.log(combinedArray);  // Outputs: [1, 2, 3, 4, 5, 6]
```

Extending the Objects

- Extend the power to objects by merging object properties effortlessly.
- Avoid convoluted code with object.assign.

JS

```
const originalObject = { name: "TechWizard", role: "Creator" };
const extendedObject = { ...originalObject, status: "Active" };

console.log(extendedObject);
// Outputs: { name: "TechWizard", role: "Creator", status: "Active"
```

Dynamic Function Arguments

- Pass dynamic arguments to functions with ease.
- Avoid the limitations of ‘apply ’and ‘arguments’.

JS

```
function displayInfo(...info) {
  console.log(info.join(", "));
}

displayInfo("Tech", "Content", "Creator");
// Outputs: Tech, Content, Creator
```

Breaking Immutability

- Destroy immutability in your code by leveraging the spread operator.
- Create modified copies without altering the original data.

JS

```
const user = { name: "John", age: 30 };
const updatedUser = { ...user, age: 31 };

console.log(updatedUser);
// Outputs: { name: "John", age: 31 }
```

DID YOU FIND IT HELPFUL ?

Share this with a friend who needs it!

VINCENT RAJA

FULL STACK WEB DEVELOPER