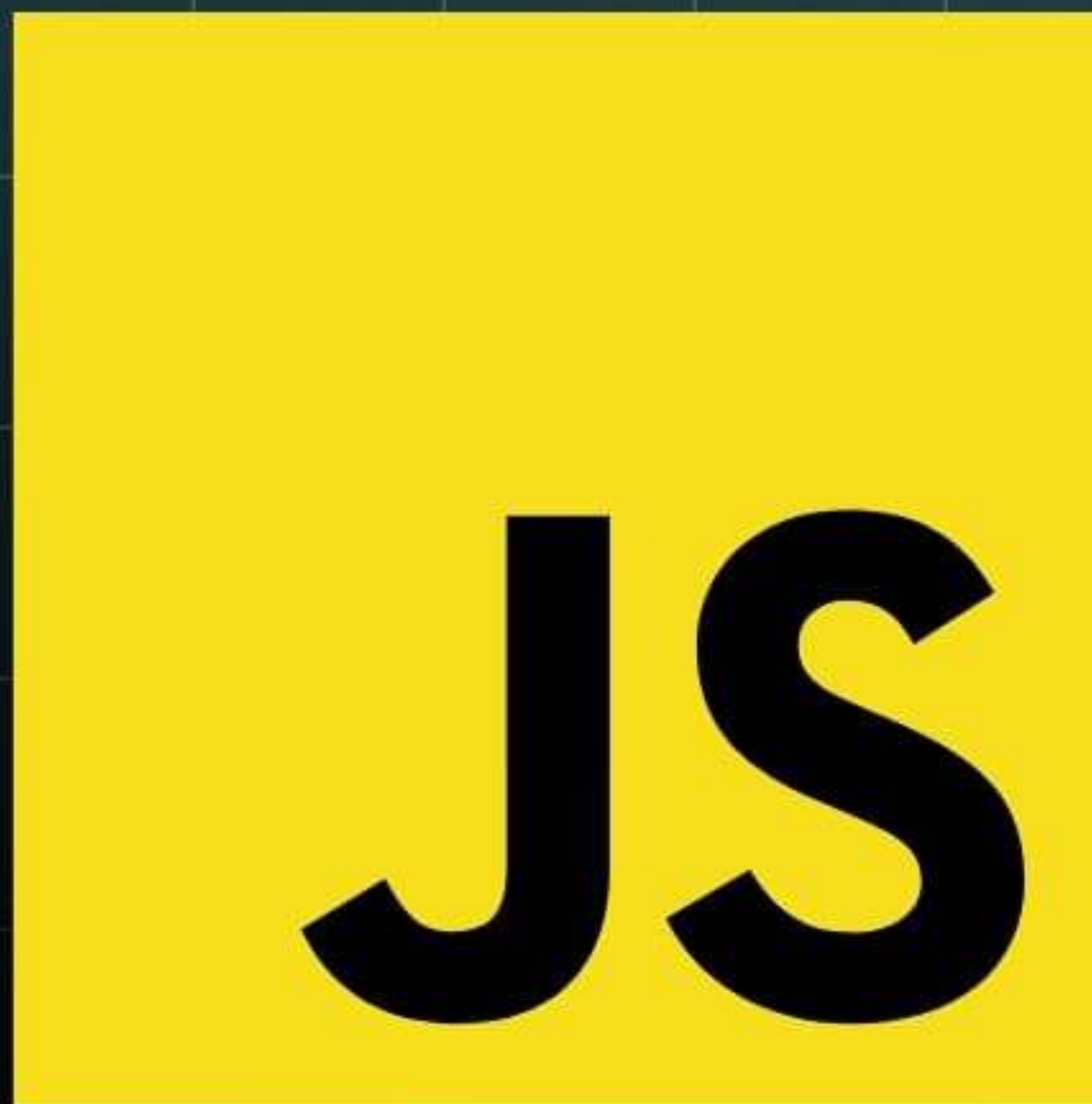




Happy Shergill
@happy.frontend

7 JAVASCRIPT CONCEPTS YOU CAN'T MISS



SAVE
THE POST

FOLLOW FOR MORE

CLOSURES

» Functions retain access to variables from their lexical scope. Facilitates modular and efficient code.



JS script.js

```
1 function outer() {  
2     let count = 0;  
3     return function inner() {  
4         count++;  
5         return count;  
6     };  
7 }  
8 const increment = outer();  
9 console.log(increment()); // Output: 1
```



PROTOTYPAL INHERITANCE

»» Objects inheriting properties and methods from other objects.

```
JS script.js

1 function Animal(name) {
2     this.name = name;
3 }
4 Animal.prototype.sound = function() {
5     return "Sound";
6 };
7 function Dog(name) {
8     Animal.call(this, name);
9 }
10 Dog.prototype = Object.create(Animal.prototype);
11 const dog = new Dog("Buddy");
12 console.log(dog.sound()); // Output: Sound
13
```



ASYNCHRONOUS PROGRAMMING

» Managing asynchronous tasks without blocking the main thread.



JS script.js

```
1 fetch('https://api.example.com/data')  
2   .then(response => response.json())  
3   .then(data => console.log(data))  
4   .catch(error => console.error(error));
```



SCOPE & HOISTING

» Variable visibility and declaration lifting.
Predictable variable behaviour and
avoiding bugs.



JS script.js

```
1 console.log(foo);  
2 // Output: undefined  
3 var foo = 'bar';  
4
```



ES6+ FEATURES

» Arrow functions, destructuring, template literals, classes, let, const.



JS script.js

```
1  const greet = (name) =>
2    `Hello, ${name}!`;
3  console.log(greet('Alice'));
4  // Output: Hello, Alice!
5
```

