Happy Shergill
@happy.frontend

7 JAVASCRIPT CONCEPTS YOU CAN'T MISS

JS

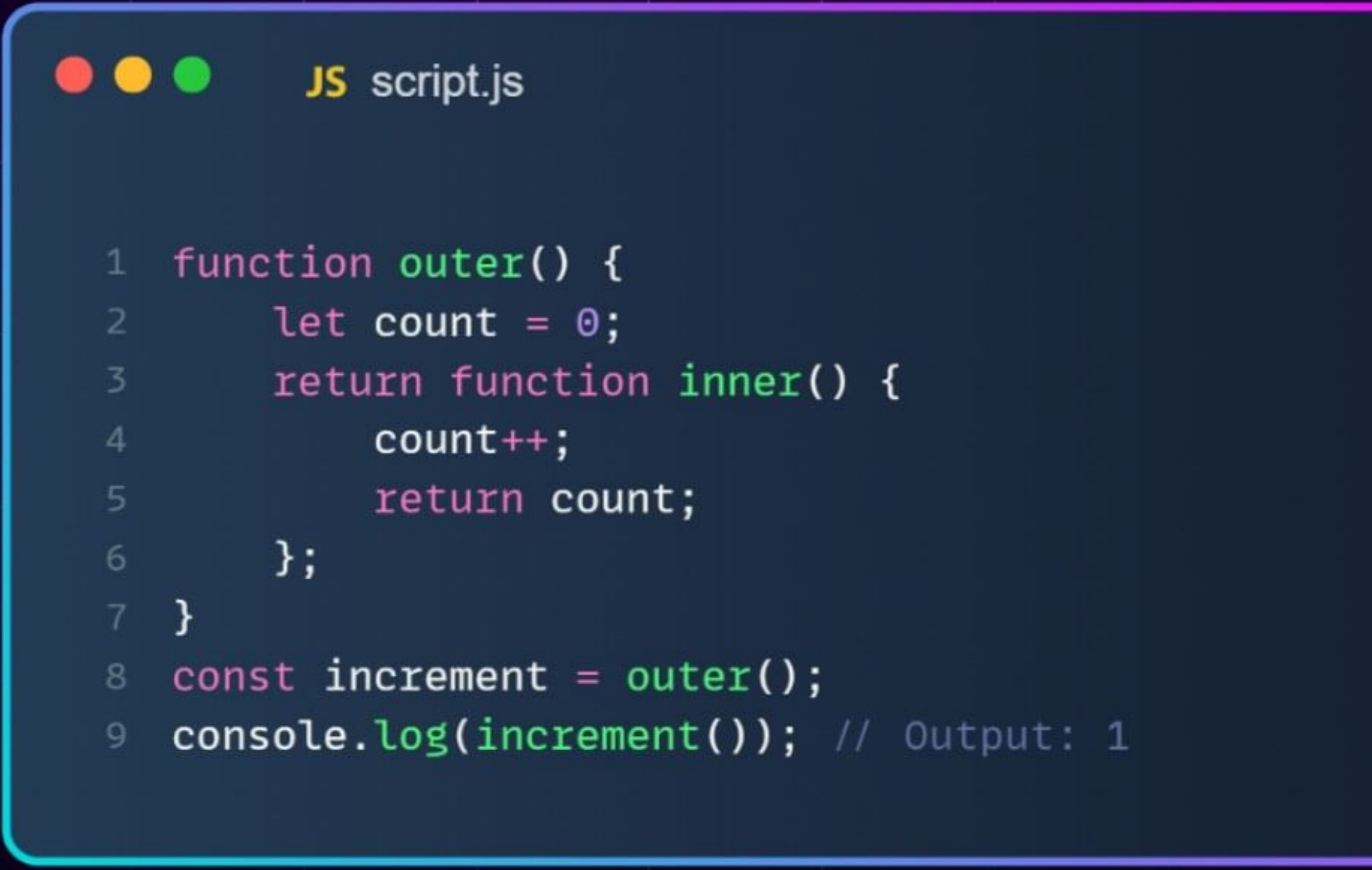
SAVE
THE POST

FOLLOW FOR MORE

SWIPE

CLOSURES

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


JS script.js

```javascript
function outer() {
    let count = 0;
    return function inner() {
        count++;
        return count;
    };
}
const increment = outer();
console.log(increment()); // Output: 1
```

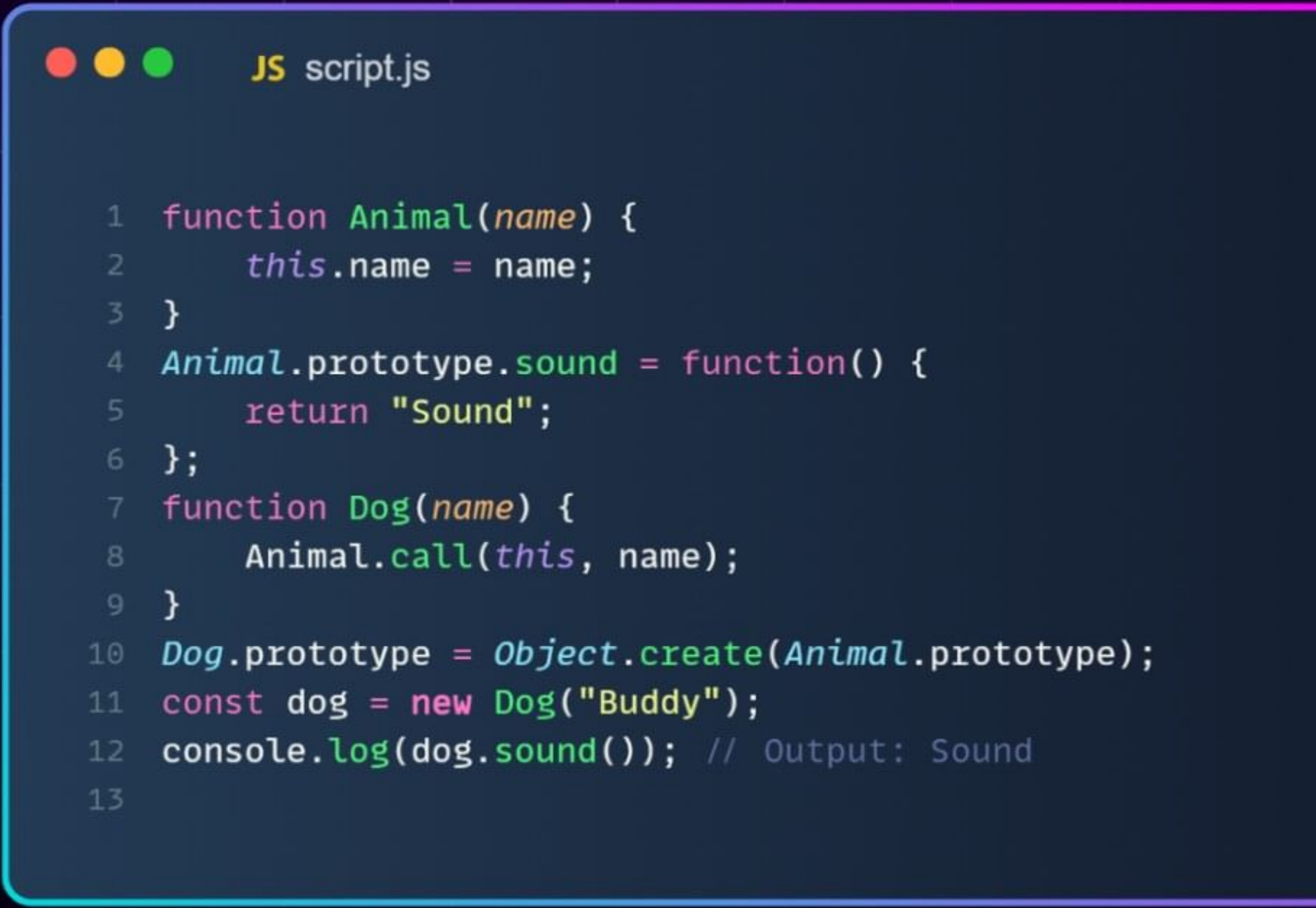
Happy Shergill
@happy.frontend

Save for later

SWIPE

PROTOTYPAL INHERITANCE

» Objects inheriting properties and methods from other objects.


JS script.js

```javascript
function Animal(name) {
    this.name = name;
}
Animal.prototype.sound = function() {
    return "Sound";
};
function Dog(name) {
    Animal.call(this, name);
}
Dog.prototype = Object.create(Animal.prototype);
const dog = new Dog("Buddy");
console.log(dog.sound()); // Output: Sound
```

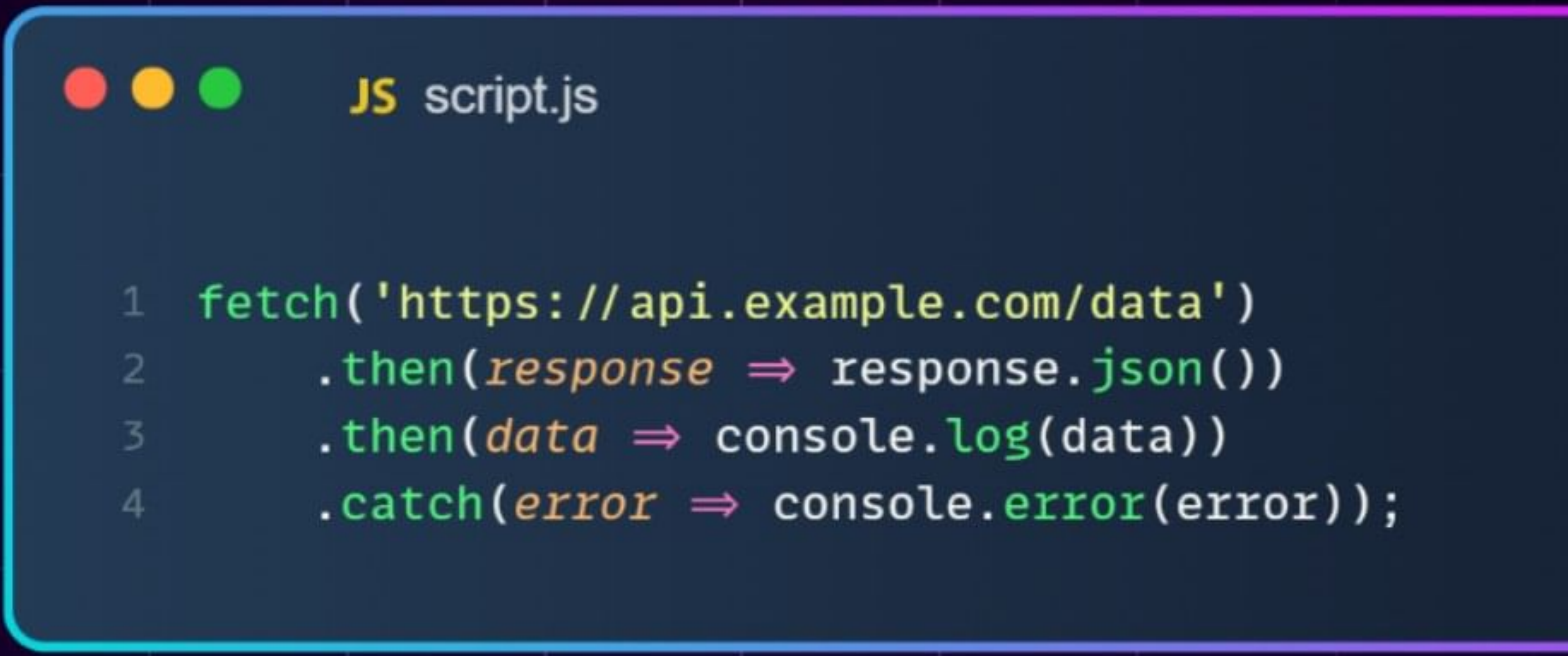
Happy Shergill
@happy.frontend

Save for later

SWIPE

ASYNCHRONOUS PROGRAMMING

Managing asynchronous tasks without blocking the main thread.



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

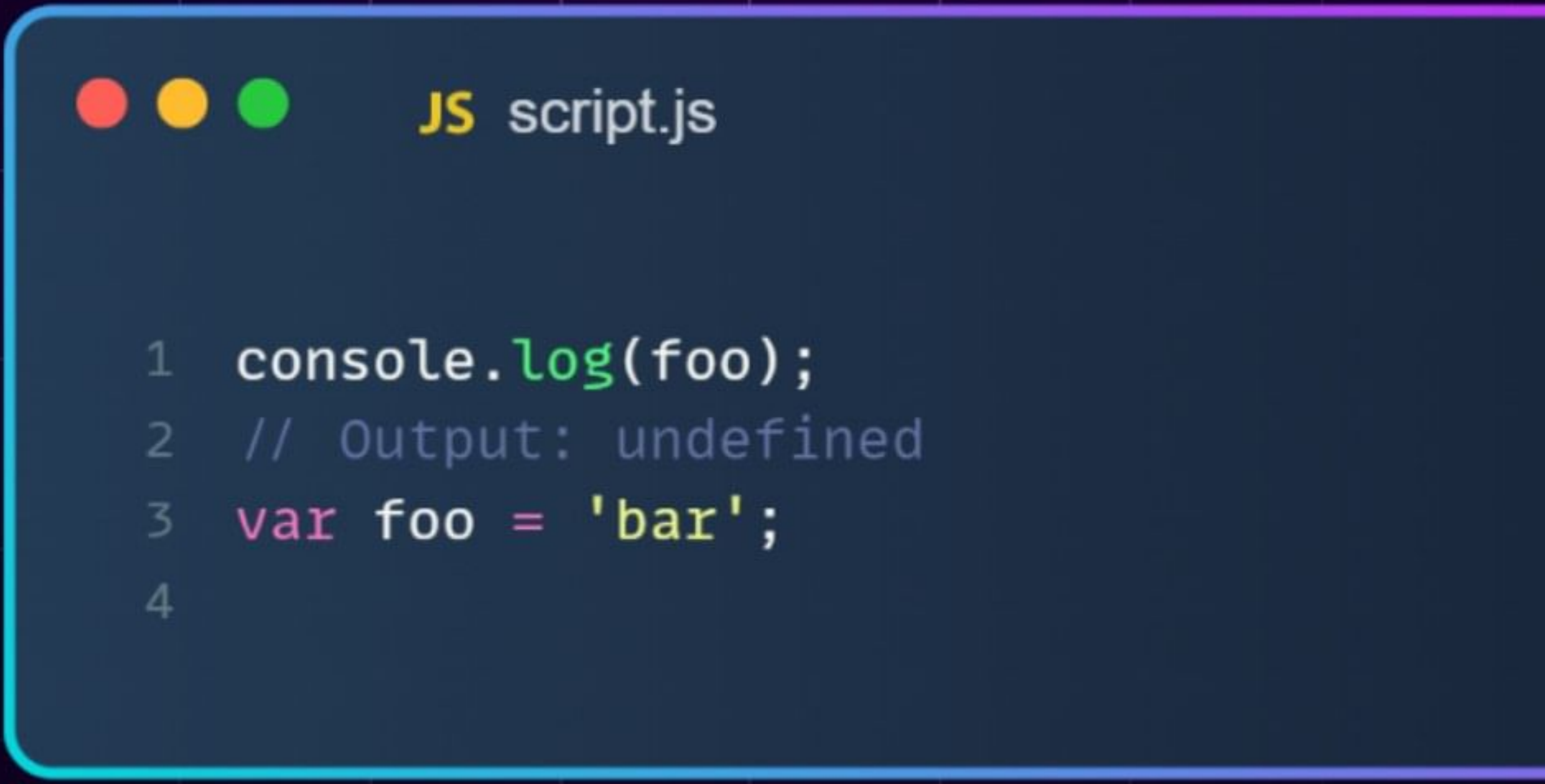
Happy Shergill
@happy.frontend

Save for later

SWIPE

SCOPE & HOISTING

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


JS script.js

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

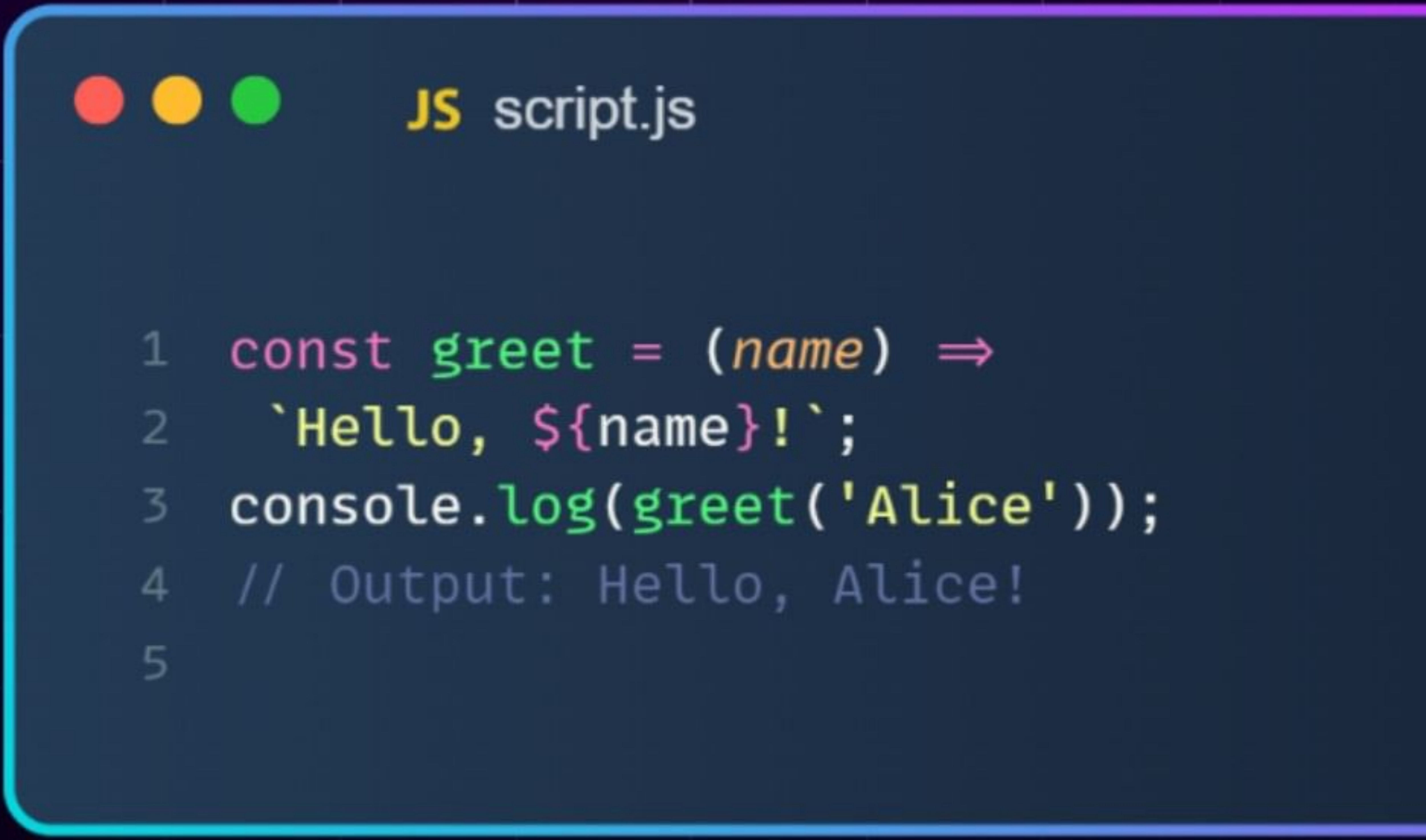
Happy Shergill
@happy.frontend

Save for later

SWIPE

ES6+ FEATURES

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



JS script.js

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

```

Happy Shergill
@happy.frontend

Save for later