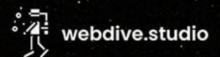
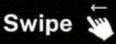


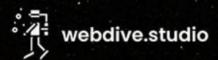
Advanced JavaScript Cheatsheet

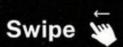




Closures

```
function createCounter() {
        let count = 0;
        return function() {
            count++;
            return count;
        };
    let counter1 = createCounter();
10
    let counter2 = createCounter();
11
12
13
    console.log(counter1()); // Output: 1
14
    console.log(counter1()); // Output: 2
15
    console.log(counter2()); // Output: 1
16
17
```





Getters and Setters

```
class Student {
        constructor(name, age) {
            this._name = name;
            this._age = age;
        }
        get name() {
            return this._name;
        set name(newName) {
11
12
            this._name = newName;
        }
        get age() {
            return this._age;
        }
        set age(newAge) {
            this._age = newAge;
    const student = new Student("Rodrigo", 28);
    console.log(student.name); // Output: "Rodrigo"
    console.log(student.age); // Output: 28
    student.name = "Alexa";
    student.age = 20;
    console.log(student.name); // Output: "Alexa"
    console.log(student.age); // Output: 20
```





Async Await

```
async function fetchData(url) {

try {

const response = await fetch(url);
 const data = await response.json();

return data;

} console.error(error);
}

async function run() {

console.log("Before fetching data");

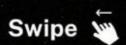
console.log("After fetching data");

console.log("After fetching data");

run();

run();
```

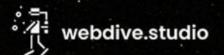


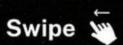


Template Literals

```
const name = "Rodrigo";
const age = 28;

const message = `Hello, my name is ${name} and I am ${age} years old.`;
```

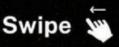




Promise

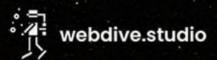
```
const fetchData = () => {
         return fetch("https://api.example.com/data")
             .then(response => {
                 if (!response.ok) {
                     throw new Error(response.statusText);
                 return response.json();
             })
             .catch(error => {
                 console.error(`Promise rejected:`, error);
10
             });
11
12
    };
13
    fetchData()
14
         .then(data => {
15
             console.log(`Promise resolved:`, data);
16
         })
17
         .catch(error => {
18
             console.error(`Promise rejected:`, error);
19
        });
20
21
```

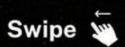




Async/Await

```
async function fetchData(url) {
        try {
            const response = await fetch(url);
            const data = await response.json();
            return data;
        } catch (error) {
11
            console.error(error);
    }
    async function run() {
        console.log("Before fetching data");
        const data = await fetchData("https://jsonplaceholder.typicode.com/posts");
20
        console.log("After fetching data");
    run();
25
```





Meta Programming

```
const person = {};
    Object.defineProperty(person, "name", {
      get: function() {
         return this._name;
      },
 6
      set: function(value) {
         this._name = value.toUpperCase();
 8
 9
    });
10
11
    person.name = "Rodrigo";
12
    console.log(person.name); // "Rodrigo"
13
14
```

