



Advance cheat Sheet

This is not an exhaustive list!



Hasibul Islam
devhasibulsilam

Closures and Scope

```
function outer() {  
  let outerVar = 'I am outer!';  
  
  function inner() {  
    console.log(outerVar);  
  }  
  return inner;  
}  
  
const closureExample = outer();  
closureExample(); // Output: I am outer!
```

LocalStorage and SessionStorage

```
// LocalStorage  
localStorage.setItem('key', 'value');  
const storedValue = localStorage.getItem('key');  
console.log(storedValue); // Output: value  
  
// SessionStorage  
sessionStorage.setItem('user', 'John');  
const user = sessionStorage.getItem('user');  
console.log(user); // Output: John
```



Hasibul Islam
devhasibulsilam

ES6 Destructuring

```
const person = { name: 'John', age: 30 };

const { name, age } = person;
console.log(name, age); // Output: John 30
```

Arrow Functions

```
const add = (a, b) => a + b;
console.log(add(3, 5)); // Output: 8
```

Promises

```
const myPromise = new Promise((resolve, reject) => {
  // Async operation
  const success = true;

  if (success) {
    resolve('Operation completed successfully!');
  } else {
    reject('Operation failed!');
  }
});

myPromise
  .then(result => console.log(result))
  .catch(error => console.error(error));
```



Hasibul Islam
devhasibulsilam

Modules

```
// math.js
export const add = (a, b) => a + b;

// main.js
import { add } from './math';
console.log(add(3, 5)); // Output: 8
```

Template Literals

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



Hasibul Islam
devhasibulsilam

Reflect

```
const obj = { name: 'John' };

// Using Reflect to get property value
console.log(Reflect.get(obj, 'name')); // Output: John

// Using Reflect to set property value
Reflect.set(obj, 'age', 30);
console.log(obj.age); // Output: 30
```

Event Delegation

```
document.addEventListener('click', function(event) {
  if (event.target.tagName === 'BUTTON') {
    console.log('Button clicked!');
  }
});
```



Hasibul Islam
devhasibulsilam

Map, Set, WeakMap, WeakSet

```
// Map
const myMap = new Map();
myMap.set('key', 'value');
console.log(myMap.get('key')) // Output: value

// Set
const mySet = new Set([1, 2, 3, 4, 5]);
console.log(mySet.has(3)); // Output: true
```

Proxy

```
const handler = {
  get: function(target, prop) {
    return prop in target ? target[prop] : 'Property not found!';
  }
};

const proxyObj = new Proxy({ name: 'John' }, handler);
console.log(proxyObj.name); // Output: John
console.log(proxyObj.age); // Output: Property not found!
```



Hasibul Islam
devhasibulsilam

Generators

```
function* generatorExample() {  
    yield 1;  
    yield 2;  
    yield 3;  
}  
  
const gen = generatorExample();  
console.log(gen.next().value); // Output: 1  
console.log(gen.next().value); // Output: 2
```

Async/Await

```
async function fetchData() {  
    try {  
        let response = await fetch('https://api.example.com/data');  
        let data = await response.json();  
        console.log(data);  
    } catch (error) {  
        console.error('Error fetching data:', error);  
    }  
}  
  
fetchData();
```



Hasibul Islam
devhasibulsilam

Classes

```
class Animal {  
  constructor(name) {  
    this.name = name;  
  }  
  
  speak() {  
    console.log(`${this.name} makes a sound.`);  
  }  
}  
  
const dog = new Animal('Dog');  
dog.speak(); // Output: Dog makes a sound.
```

Rest and Spread Operators

```
// Rest Operator  
function sum(...numbers) {  
  return numbers.reduce((total, num) => total + num, 0);  
}  
  
console.log(sum(1, 2, 3, 4)); // Output: 10  
  
// Spread Operator  
const arr1 = [1, 2, 3];  
const arr2 = [...arr1, 4, 5];  
console.log(arr2); // Output: [1, 2, 3, 4, 5]
```



Hasibul Islam

@devhasibulislam

Did you find it useful?
Share your thoughts.



Type on social

devhasibulislam



Follow me