JavaScript Complete Notes

■ Rest Operator

The Rest operator (...) collects multiple values into a single array or object.

■ *In Functions:*

```
function sum(...numbers) { return numbers.reduce((total, n) => total + n, 0); }
console.log(sum(1, 2, 3, 4)); // 10
```

■ *In Destructuring:*

```
let person = { name: "Ali", age: 25, city: "Karachi" }; let { name, ...other } =
person; console.log(name); // Ali console.log(other); // { age: 25, city:
"Karachi" }
```

■ Spread Operator

The Spread operator (...) expands values from an array or object.

■ *With Arrays:*

```
let arr1 = [1, 2]; let arr2 = [3, 4]; let combined = [...arr1, ...arr2];
console.log(combined); // [1, 2, 3, 4]
```

■ *With Functions:*

```
function greet(a, b, c) { console.log(a, b, c); } let args = ["Hello", "World",
"!"]; greet(...args); // Hello World !
```

■ *With Objects:*

```
let obj1 = { name: "Aqsa" }; let obj2 = { age: 22 }; let user = { ...obj1, ...obj2
}; console.log(user); // { name: "Aqsa", age: 22 }
```

■ Promises

A Promise represents the eventual completion (or failure) of an asynchronous task.

■ *Basic Example:*

```
let myPromise = new Promise((resolve, reject) => { let success = true; if
(success) { resolve("Task Done ■"); } else { reject("Task Failed ■"); } });
myPromise .then(res => console.log(res)) .catch(err => console.log(err));
```

■ *Example with setTimeout (Async):*

```
let promiseExample = new Promise((resolve) => { setTimeout(() => { resolve("Data
Loaded ■"); }, 2000); }); promiseExample.then(msg => console.log(msg));
```

■ Synchronous vs Asynchronous

■ *Synchronous Example:*

```
console.log("Task 1"); console.log("Task 2"); console.log("Task 3"); // Output:
// Task 1 // Task 2 // Task 3
```

■ *Asynchronous Example:*

```
console.log("Task 1"); setTimeout(() => { console.log("Task 2 (after 2s)"); },
2000); console.log("Task 3"); // Output: // Task 1 // Task 3 // Task 2 (after 2s)
```

■ Regular Expressions (Regex)

Regular Expressions are patterns used to match text.

■ *Example 1: Test if a string contains 'cat'*

```
let regex = /cat/; console.log(regex.test("I have a cat")); // true
console.log(regex.test("I have a dog")); // false
```

■ *Example 2: Match digits*

```
let regex = /\d+/; console.log("My age is 25".match(regex)); // ["25"]
```

■ *Example 3: Validate Email*

```
let emailRegex = /^[\w.-]+@[\w.-]+\.\w+$/;
console.log(emailRegex.test("test@example.com")); // true
console.log(emailRegex.test("invalid@com")); // false
```