**javascript**

**Easy level**

**JavaScript Basics**

1. **What is the correct syntax to output "Hello World" in JavaScript?**  
   A. print("Hello World")  
   B. console.log("Hello World")  
   C. System.out.println("Hello World")  
   D. echo("Hello World")  
   **Answer:** B
2. **Which data type is NOT supported in JavaScript?**  
   A. String  
   B. Number  
   C. Float  
   D. Boolean  
   **Answer:** C
3. **Which company developed JavaScript?**  
   A. Microsoft  
   B. Google  
   C. Netscape  
   D. Apple  
   **Answer:** C
4. **How do you create a function in JavaScript?**  
   A. function myFunction()  
   B. function:myFunction()  
   C. def myFunction()  
   D. func myFunction()  
   **Answer:** A
5. **How do you call a function named myFunction?**  
   A. call myFunction()  
   B. myFunction()  
   C. call function myFunction  
   D. execute myFunction()  
   **Answer:** B

**Variables and Constants**

1. **Which keyword is used to declare a variable in JavaScript?**  
   A. var  
   B. let  
   C. const  
   D. All of the above  
   **Answer:** D
2. **What is the correct way to declare a constant?**  
   A. constant x = 10;  
   B. const x = 10;  
   C. var x = 10;  
   D. let x = 10;  
   **Answer:** B
3. **What is the default value of an uninitialized variable in JavaScript?**  
   A. undefined  
   B. null  
   C. 0  
   D. NaN  
   **Answer:** A

**Operators**

1. **Which operator is used to assign a value to a variable?**  
   A. =  
   B. ==  
   C. ===  
   D. =>  
   **Answer:** A
2. **What does === mean in JavaScript?**  
   A. Assignment operator  
   B. Equality without type conversion  
   C. Equality with type conversion  
   D. None of the above  
   **Answer:** B

**Loops**

1. **Which loop is guaranteed to run at least once?**  
   A. for  
   B. while  
   C. do-while  
   D. None of the above  
   **Answer:** C
2. **How do you stop a loop in JavaScript?**  
   A. exit  
   B. break  
   C. return  
   D. continue  
   **Answer:** B

**Functions**

1. **What does the return keyword do in a function?**  
   A. Stops execution of the function  
   B. Returns a value from the function  
   C. Both A and B  
   D. None of the above  
   **Answer:** C
2. **What is a callback function?**  
   A. A function that calls itself  
   B. A function passed as an argument to another function  
   C. A function that runs after the main function  
   D. None of the above  
   **Answer:** B

**Arrays**

1. **How do you create an array in JavaScript?**  
   A. let arr = {};  
   B. let arr = [];  
   C. let arr = ();  
   D. let arr = //;  
   **Answer:** B
2. **What is the method used to add elements to the end of an array?**  
   A. push()  
   B. pop()  
   C. shift()  
   D. unshift()  
   **Answer:** A
3. **Which method removes the last element from an array?**  
   A. remove()  
   B. pop()  
   C. delete()  
   D. splice()  
   **Answer:** B

**Objects**

1. **How do you define an object in JavaScript?**  
   A. let obj = {};  
   B. let obj = [];  
   C. let obj = ();  
   D. let obj = <>;  
   **Answer:** A
2. **How do you access a property in an object?**  
   A. obj.property  
   B. obj[property]  
   C. Both A and B  
   D. None of the above  
   **Answer:** C

**Events**

1. **Which event occurs when a user clicks on an HTML element?**  
   A. onmouseclick  
   B. onclick  
   C. onhover  
   D. onchange  
   **Answer:** B
2. **How do you write an event listener in JavaScript?**  
   A. element.addEventListener("click", myFunction);  
   B. element.listen("click", myFunction);  
   C. element.on("click", myFunction);  
   D. element.trigger("click", myFunction);  
   **Answer:** A

**DOM Manipulation**

1. **Which method selects an element by ID?**  
   A. getElementById()  
   B. querySelector()  
   C. getElementsByClassName()  
   D. getElementsByTagName()  
   **Answer:** A
2. **How do you change the content of an HTML element using JavaScript?**  
   A. element.innerHTML = "new content";  
   B. element.value = "new content";  
   C. element.textContent = "new content";  
   D. All of the above  
   **Answer:** D

**Strings**

1. **How do you find the length of a string in JavaScript?**  
   A. string.size  
   B. string.length  
   C. string.count  
   D. string.len  
   **Answer:** B
2. **What method is used to convert a string to uppercase?**  
   A. toUpperCase()  
   B. toUpper()  
   C. upperCase()  
   D. convertToUpper()  
   **Answer:** A
3. **Which method is used to split a string into an array?**  
   A. slice()  
   B. split()  
   C. splice()  
   D. join()  
   **Answer:** B
4. **How do you find the index of a substring in a string?**  
   A. string.search()  
   B. string.indexOf()  
   C. string.substr()  
   D. string.position()  
   **Answer:** B

**Type Conversion**

1. **Which method converts a string to an integer?**  
   A. parseFloat()  
   B. Number()  
   C. parseInt()  
   D. String()  
   **Answer:** C
2. **How do you convert a number to a string in JavaScript?**  
   A. toString()  
   B. String()  
   C. Both A and B  
   D. None of the above  
   **Answer:** C

**Date and Time**

1. **Which object is used to work with dates and times in JavaScript?**  
   A. Time  
   B. Clock  
   C. Date  
   D. Calendar  
   **Answer:** C
2. **How do you get the current year using the Date object?**  
   A. Date.getYear()  
   B. new Date().getYear()  
   C. new Date().getFullYear()  
   D. Date.now()  
   **Answer:** C
3. **What method returns the current time in milliseconds since January 1, 1970?**  
   A. Date.getMilliseconds()  
   B. Date.now()  
   C. Date.getTime()  
   D. Date.milliseconds()  
   **Answer:** B

**Math**

1. **Which method rounds a number to the nearest integer?**  
   A. Math.floor()  
   B. Math.ceil()  
   C. Math.round()  
   D. Math.abs()  
   **Answer:** C
2. **How do you generate a random number in JavaScript?**  
   A. Math.random()  
   B. Math.randomInt()  
   C. Math.rand()  
   D. Math.randomize()  
   **Answer:** A
3. **What is the result of Math.floor(4.7)?**  
   A. 4  
   B. 5  
   C. 4.7  
   D. 0  
   **Answer:** A

**Control Statements**

1. **What is the output of if (false) { console.log("Hello"); } else { console.log("World"); }?**  
   A. Hello  
   B. World  
   C. Hello World  
   D. None  
   **Answer:** B
2. **What does the switch statement do in JavaScript?**  
   A. Executes a single block of code  
   B. Executes multiple blocks of code  
   C. Evaluates an expression and matches it to a case block  
   D. Stops the script  
   **Answer:** C

**Error Handling**

1. **Which statement is used to handle errors in JavaScript?**  
   A. try-catch  
   B. try-except  
   C. error-catch  
   D. catch-finally  
   **Answer:** A
2. **What does the finally block do in a try-catch-finally statement?**  
   A. Executes only if an error occurs  
   B. Executes only if no error occurs  
   C. Always executes after the try and catch blocks  
   D. None of the above  
   **Answer:** C

**JSON**

1. **What does JSON stand for?**  
   A. Java Standard Object Notation  
   B. JavaScript Object Notation  
   C. Java Syntax Object Notation  
   D. None of the above  
   **Answer:** B
2. **How do you parse a JSON string into a JavaScript object?**  
   A. JSON.parse()  
   B. JSON.stringify()  
   C. JSON.convert()  
   D. JSON.toObject()  
   **Answer:** A
3. **Which method converts a JavaScript object to a JSON string?**  
   A. JSON.stringify()  
   B. JSON.parse()  
   C. JSON.encode()  
   D. JSON.toString()  
   **Answer:** A

**Advanced Topics**

1. **What is a promise in JavaScript?**  
   A. A function that runs immediately  
   B. An object representing the eventual completion or failure of an asynchronous operation  
   C. A function that resolves instantly  
   D. A synchronous operation  
   **Answer:** B
2. **Which method is used to attach a callback to a promise?**  
   A. .call()  
   B. .then()  
   C. .catch()  
   D. .resolve()  
   **Answer:** B
3. **What is an arrow function in JavaScript?**  
   A. A function declared with function keyword  
   B. A shorthand for declaring functions using =>  
   C. A synchronous function  
   D. None of the above  
   **Answer:** B

**Medium level**

**1. Arrays**

1. **What is the output of [1, 2, 3].map(x => x \* 2)?**  
   A. [1, 2, 3]  
   B. [2, 4, 6]  
   C. [1, 4, 9]  
   D. Error  
   **Answer:** B
2. **How do you remove the last element from an array?**  
   A. array.pop()  
   B. array.shift()  
   C. array.splice(-1)  
   D. array.remove()  
   **Answer:** A
3. **What does Array.prototype.filter() do?**  
   A. Modifies the array  
   B. Creates a new array with elements that pass a test  
   C. Reduces the array to a single value  
   D. Sorts the array  
   **Answer:** B
4. **How do you merge two arrays in JavaScript?**  
   A. array1.concat(array2)  
   B. array1.join(array2)  
   C. array1.add(array2)  
   D. None of the above  
   **Answer:** A

**2. Functions**

1. **What is a higher-order function in JavaScript?**  
   A. A function that takes another function as an argument or returns a function  
   B. A function that runs immediately  
   C. A function declared globally  
   D. None of the above  
   **Answer:** A
2. **What is the default value of a function parameter in JavaScript?**  
   A. undefined  
   B. null  
   C. 0  
   D. Empty string  
   **Answer:** A
3. **What is the output of this code?**

javascript

CopyEdit

function greet(name = "Guest") {

return `Hello, ${name}`;

}

console.log(greet());

A. Hello, Guest  
B. Hello, undefined  
C. Hello, null  
D. Error  
**Answer:** A

1. **What is the use of the apply() method?**  
   A. Calls a function with a specified this value and arguments as an array  
   B. Creates a new function  
   C. Immediately executes a function  
   D. None of the above  
   **Answer:** A

**3. Objects**

1. **How do you access a property of an object using a variable?**  
   A. object.property  
   B. object[variable]  
   C. object{variable}  
   D. object(variable)  
   **Answer:** B
2. **Which method is used to copy object properties into another object?**  
   A. Object.create()  
   B. Object.assign()  
   C. Object.copy()  
   D. Object.extend()  
   **Answer:** B
3. **What does the Object.keys() method return?**  
   A. An array of key-value pairs  
   B. An array of the object's keys  
   C. The object's values  
   D. The object's properties  
   **Answer:** B
4. **What is the result of typeof null in JavaScript?**  
   A. "null"  
   B. "object"  
   C. "undefined"  
   D. "string"  
   **Answer:** B

**4. Closures**

1. **What is a closure in JavaScript?**  
   A. A function bundled with its lexical environment  
   B. A function executed immediately  
   C. A private variable  
   D. None of the above  
   **Answer:** A
2. **What will the following code output?**

javascript

CopyEdit

function outer() {

let count = 0;

return function inner() {

count++;

return count;

};

}

const counter = outer();

console.log(counter());

console.log(counter());

A. 0, 0  
B. 1, 2  
C. 1, 1  
D. Error  
**Answer:** B

**5. Promises and Async/Await**

1. **Which method is used to handle a rejected promise?**  
   A. then()  
   B. reject()  
   C. catch()  
   D. finally()  
   **Answer:** C
2. **What is the purpose of async functions?**  
   A. To make a function execute immediately  
   B. To make a function execute synchronously  
   C. To handle asynchronous code more easily with await  
   D. To stop the execution of the program  
   **Answer:** C
3. **What will the following code output?**

javascript

CopyEdit

async function example() {

return "Hello";

}

example().then(console.log);

A. Hello  
B. Promise {<resolved>: "Hello"}  
C. Error  
D. None of the above  
**Answer:** A

**6. DOM Manipulation**

1. **Which method selects the first matching element in the DOM?**  
   A. getElementById()  
   B. querySelector()  
   C. getElementsByClassName()  
   D. querySelectorAll()  
   **Answer:** B
2. **How do you add a new element to the DOM?**  
   A. document.addElement()  
   B. document.createElement()  
   C. document.appendChild()  
   D. Both B and C  
   **Answer:** D
3. **What does addEventListener() do in JavaScript?**  
   A. Adds an event to an element  
   B. Executes an event immediately  
   C. Attaches a handler to an event  
   D. Both A and C  
   **Answer:** D

**7. Scope**

1. **What is the difference between var, let, and const?**  
   A. var is function-scoped; let and const are block-scoped  
   B. var is block-scoped; let and const are function-scoped  
   C. const is mutable  
   D. No difference  
   **Answer:** A
2. **What is the output of the following code?**

javascript

CopyEdit

var a = 5;

function test() {

var a = 10;

console.log(a);

}

test();

console.log(a);

A. 10, 10  
B. 5, 5  
C. 10, 5  
D. Error  
**Answer:** C

Would you like me to continue or focus on specific **advanced concepts** (e.g., **event loops**, **ES6 features**, or **testing in JavaScript**) for the rest?

**You said:**

yes

**ChatGPT said:**

ChatGPT

**Advanced JavaScript Questions (Medium Difficulty)**

**8. Event Loop and Asynchronous JavaScript**

1. **What is the event loop in JavaScript?**  
   A. A process that handles asynchronous tasks in JavaScript  
   B. A function that executes immediately  
   C. A queue of synchronous functions  
   D. A debugging tool  
   **Answer:** A
2. **What is the output of the following code?**

javascript

CopyEdit

console.log("Start");

setTimeout(() => console.log("Timeout"), 0);

console.log("End");

A. Start Timeout End  
B. Start End Timeout  
C. Timeout Start End  
D. End Start Timeout  
**Answer:** B

1. **Which of the following is NOT a phase in the event loop?**  
   A. Timer Phase  
   B. Callback Phase  
   C. Poll Phase  
   D. Synchronous Phase  
   **Answer:** D
2. **What does Promise.resolve() do?**  
   A. Rejects a promise  
   B. Immediately resolves a promise  
   C. Delays the promise  
   D. Creates a new rejected promise  
   **Answer:** B

**9. ES6 Features**

1. **What is the purpose of the let keyword in JavaScript?**  
   A. To declare a constant variable  
   B. To declare block-scoped variables  
   C. To declare variables globally  
   D. None of the above  
   **Answer:** B
2. **What is the difference between == and ===?**  
   A. == compares both value and type, === compares only value  
   B. == compares only value, === compares both value and type  
   C. Both are the same  
   D. None of the above  
   **Answer:** B
3. **What is the purpose of the ... operator in JavaScript?**  
   A. To merge arrays or objects  
   B. To destructure arrays or objects  
   C. Both A and B  
   D. None of the above  
   **Answer:** C
4. **What does const do in JavaScript?**  
   A. Declares a variable that cannot be reassigned or mutated  
   B. Declares a block-scoped variable that cannot be reassigned  
   C. Declares a variable globally  
   D. None of the above  
   **Answer:** B

**10. Error Handling**

1. **What is the purpose of a try...catch block?**  
   A. To debug code  
   B. To handle errors in a block of code  
   C. To immediately execute a function  
   D. None of the above  
   **Answer:** B
2. **What is the output of the following code?**

javascript

CopyEdit

try {

console.log("Try block");

throw new Error("Error occurred");

} catch (err) {

console.log("Catch block");

} finally {

console.log("Finally block");

}

A. Try block Catch block  
B. Try block Finally block  
C. Try block Catch block Finally block  
D. Error  
**Answer:** C

1. **Which method is used to manually throw an error?**  
   A. throw new Error()  
   B. throw.error()  
   C. console.error()  
   D. None of the above  
   **Answer:** A

**11. Classes**

1. **What is a class in JavaScript?**  
   A. A blueprint for creating objects  
   B. A function that executes immediately  
   C. A block-scoped variable  
   D. None of the above  
   **Answer:** A
2. **What is the purpose of the constructor method in a class?**  
   A. To define methods in a class  
   B. To initialize object properties  
   C. To create static methods  
   D. None of the above  
   **Answer:** B
3. **What will the following code output?**

javascript

CopyEdit

class Animal {

constructor(name) {

this.name = name;

}

speak() {

return `${this.name} makes a noise.`;

}

}

const dog = new Animal("Dog");

console.log(dog.speak());

A. Dog speaks.  
B. Dog makes a noise.  
C. undefined makes a noise.  
D. Error  
**Answer:** B

1. **What is the purpose of the static keyword in a class?**  
   A. To create methods shared across all instances of the class  
   B. To define private properties  
   C. To inherit from another class  
   D. To create immutable properties  
   **Answer:** A

**12. Modules**

1. **How do you import a module in JavaScript?**  
   A. import moduleName from "module"  
   B. require("module")  
   C. include("module")  
   D. Both A and B  
   **Answer:** D
2. **What does the export default keyword do?**  
   A. Exports all methods in a file  
   B. Exports a single default function, object, or variable  
   C. Exports multiple functions or objects  
   D. None of the above  
   **Answer:** B
3. **Which statement is true about named exports?**  
   A. They allow you to export multiple values from a file  
   B. They allow you to import values using any name  
   C. Both A and B  
   D. None of the above  
   **Answer:** A

**13. Advanced Topics**

1. **What does Object.freeze() do?**  
   A. Prevents properties from being added to an object  
   B. Prevents properties from being modified in an object  
   C. Both A and B  
   D. None of the above  
   **Answer:** C
2. **What is the output of the following code?**

javascript

CopyEdit

const obj = { name: "John" };

Object.freeze(obj);

obj.name = "Jane";

console.log(obj.name);

A. Jane  
B. John  
C. Error  
D. Undefined  
**Answer:** B

1. **What is a WeakMap in JavaScript?**  
   A. A collection of key-value pairs where keys must be objects  
   B. A collection of key-value pairs where keys are strings  
   C. A map with weak references to its keys  
   D. Both A and C  
   **Answer:** D

**Difficult**

**1. Closures**

1. **What is a closure in JavaScript? How does it work?  
   A. A function that can access variables in its own scope only  
   B. A function that can access variables from its scope and parent scopes  
   C. A function that executes immediately  
   D. A block-scoped variable  
   Answer: B**
2. **What will the following code output? Why?**

**javascript**

**CopyEdit**

**function createCounter() {**

**let count = 0;**

**return function () {**

**count++;**

**return count;**

**};**

**}**

**const counter = createCounter();**

**console.log(counter()); // ?**

**console.log(counter()); // ?**

**A. 1, 1  
B. 0, 0  
C. 1, 2  
D. Error  
Answer: C**

1. **What happens when you use a closure inside a loop with var instead of let? How can you fix it?  
   A. The closure will use the same variable for all iterations  
   B. The closure will create a new variable for each iteration  
   C. The closure will throw an error  
   D. None of the above  
   Answer: A**

**2. Prototypes**

1. **What is the difference between \_\_proto\_\_ and prototype in JavaScript?  
   A. \_\_proto\_\_ is an object, prototype is a function  
   B. \_\_proto\_\_ is used for inheritance, prototype is used for methods  
   C. Both are the same  
   D. None of the above  
   Answer: B**
2. **What will the following code output?**

**javascript**

**CopyEdit**

**function Person(name) {**

**this.name = name;**

**}**

**Person.prototype.greet = function () {**

**return `Hello, ${this.name}`;**

**};**

**const john = new Person("John");**

**console.log(john.greet());**

**A. Hello, John  
B. Undefined  
C. Error  
D. Hello, undefined  
Answer: A**

**3. Asynchronous Programming**

1. **What will the following code output?**

**javascript**

**CopyEdit**

**async function test() {**

**return 5;**

**}**

**console.log(test());**

**A. 5  
B. Promise { <pending> }  
C. Promise { 5 }  
D. Error  
Answer: C**

1. **How does the async/await syntax differ from then/catch in promises?  
   A. async/await makes asynchronous code easier to read  
   B. then/catch allows chaining multiple promises  
   C. Both A and B  
   D. None of the above  
   Answer: C**
2. **What is the purpose of Promise.allSettled()?  
   A. Resolves when all promises resolve  
   B. Rejects when any promise rejects  
   C. Waits for all promises to either resolve or reject  
   D. None of the above  
   Answer: C**

**4. Memory Management**

1. **What is a memory leak in JavaScript?  
   A. An unused variable  
   B. A retained reference that prevents garbage collection  
   C. A reference to a null object  
   D. None of the above  
   Answer: B**
2. **How can you avoid memory leaks in JavaScript?  
   A. Use let and const instead of var  
   B. Avoid global variables  
   C. Remove event listeners when they are no longer needed  
   D. All of the above  
   Answer: D**

**5. Advanced Object Concepts**

1. **What is the output of the following code?**

**javascript**

**CopyEdit**

**const obj = {**

**a: 10,**

**b: {**

**c: 20,**

**d: function () {**

**return this.c;**

**},**

**},**

**};**

**console.log(obj.b.d());**

**A. 20  
B. undefined  
C. null  
D. Error  
Answer: A**

1. **What is the purpose of Object.defineProperty()?  
   A. To add new properties to an object with fine-grained control  
   B. To prevent modification of properties  
   C. Both A and B  
   D. None of the above  
   Answer: C**
2. **What is the output of this code? Why?**

**javascript**

**CopyEdit**

**const obj = {**

**name: "Alice",**

**getName: () => this.name,**

**};**

**console.log(obj.getName());**

**A. "Alice"  
B. undefined  
C. Error  
D. null  
Answer: B**

**6. Symbols and Iterators**

1. **What is a symbol in JavaScript?  
   A. A new primitive type introduced in ES6  
   B. A unique and immutable identifier  
   C. Both A and B  
   D. None of the above  
   Answer: C**
2. **What is the output of the following code?**

**javascript**

**CopyEdit**

**const sym1 = Symbol("id");**

**const sym2 = Symbol("id");**

**console.log(sym1 === sym2);**

**A. true  
B. false  
C. Error  
D. undefined  
Answer: B**

**7. Advanced Functionality**

1. **What will the following code output?**

**javascript**

**CopyEdit**

**const multiply = (a, b = 1) => a \* b;**

**console.log(multiply(5));**

**A. 5  
B. NaN  
C. Error  
D. undefined  
Answer: A**

1. **What is function currying in JavaScript?  
   A. A function that returns another function with arguments pre-applied  
   B. A function that modifies its input  
   C. A method of copying functions  
   D. None of the above  
   Answer: A**

**8. Modules**

1. **Why should you use ES6 modules over older require statements?  
   A. They allow tree-shaking  
   B. They are asynchronous  
   C. Both A and B  
   D. None of the above  
   Answer: C**
2. **What is the purpose of import.meta in JavaScript?  
   A. To fetch module-specific metadata  
   B. To import dynamic modules  
   C. To modify imported objects  
   D. None of the above  
   Answer: A**

**9. Event Delegation**

1. **What is event delegation in JavaScript?  
   A. The process of delegating events to parent elements  
   B. Using addEventListener to assign multiple events  
   C. Both A and B  
   D. None of the above  
   Answer: A**

**10. Advanced Scope and Context**

1. **What will the following code output? Why?**

**javascript**

**CopyEdit**

**const obj = {**

**name: "John",**

**logName() {**

**console.log(this.name);**

**},**

**};**

**const log = obj.logName;**

**log();**

**A. John  
B. undefined  
C. null  
D. Error  
Answer: B (because this in log() refers to the global object, not obj)**

1. **What will the following code output?**

**javascript**

**CopyEdit**

**(function () {**

**console.log(this);**

**})();**

**A. undefined  
B. Window (or global object in Node.js)  
C. Error  
D. null  
Answer: B**

**11. Advanced Promises**

1. **What will the following code output?**

**javascript**

**CopyEdit**

**const promise = new Promise((resolve, reject) => {**

**resolve("Success");**

**reject("Error");**

**});**

**promise.then(console.log).catch(console.error);**

**A. Success  
B. Error  
C. Both Success and Error  
D. None of the above  
Answer: A**

1. **What is the output of the following?**

**javascript**

**CopyEdit**

**console.log("Start");**

**setTimeout(() => console.log("Timeout"), 0);**

**Promise.resolve().then(() => console.log("Promise"));**

**console.log("End");**

**A. Start, End, Timeout, Promise  
B. Start, Promise, End, Timeout  
C. Start, End, Promise, Timeout  
D. Promise, Start, Timeout, End  
Answer: C**

**12. Advanced Array and Object Manipulation**

1. **What does the reduce() method do?  
   A. Iterates over an array and returns a single value  
   B. Filters elements in an array  
   C. Modifies an array in-place  
   D. None of the above  
   Answer: A**
2. **What will the following code output?**

**javascript**

**CopyEdit**

**const nums = [1, 2, 3, 4];**

**const total = nums.reduce((acc, curr) => acc + curr, 0);**

**console.log(total);**

**A. 10  
B. 0  
C. undefined  
D. Error  
Answer: A**

1. **How can you deeply clone an object in JavaScript?  
   A. Object.assign({}, obj)  
   B. JSON.parse(JSON.stringify(obj))  
   C. Both A and B  
   D. None of the above  
   Answer: B**

**13. Event Loop and Callbacks**

1. **What is the event loop in JavaScript?  
   A. A process that handles setTimeout and setInterval  
   B. A mechanism that executes callbacks asynchronously  
   C. Both A and B  
   D. None of the above  
   Answer: C**
2. **What is the output of this code?**

**javascript**

**CopyEdit**

**console.log(1);**

**setTimeout(() => console.log(2), 0);**

**console.log(3);**

**A. 1, 2, 3  
B. 1, 3, 2  
C. 2, 1, 3  
D. 3, 1, 2  
Answer: B**

**14. Advanced Error Handling**

1. **What will the following code output?**

**javascript**

**CopyEdit**

**try {**

**console.log("Try Block");**

**throw new Error("Error Block");**

**} catch (err) {**

**console.log("Catch Block");**

**} finally {**

**console.log("Finally Block");**

**}**

**A. Try Block, Catch Block, Finally Block  
B. Error Block, Finally Block  
C. Catch Block, Finally Block  
D. Try Block, Finally Block  
Answer: A**

1. **What happens if an error is thrown inside a finally block?  
   A. The error is ignored  
   B. The error replaces the original error  
   C. The original error is thrown  
   D. None of the above  
   Answer: B**

**15. Advanced Regular Expressions**

1. **What will the following code output?**

**javascript**

**CopyEdit**

**const regex = /(\d+)/g;**

**const str = "Price: 123, Discount: 45";**

**console.log(str.match(regex));**

**A. [123, 45]  
B. 123  
C. 45  
D. Error  
Answer: A**

1. **How do you perform a global search-and-replace using regex?  
   A. str.replace("pattern", "replacement")  
   B. str.replace(/pattern/g, "replacement")  
   C. str.match("pattern")  
   D. None of the above  
   Answer: B**

**16. Advanced Design Patterns**

1. **What is the Singleton pattern in JavaScript?  
   A. A function that creates unique objects every time  
   B. A function that ensures only one instance of an object exists  
   C. A function that clones an object  
   D. None of the above  
   Answer: B**
2. **What is the output of this module pattern?**

**javascript**

**CopyEdit**

**const Counter = (function () {**

**let count = 0;**

**return {**

**increment: function () {**

**count++;**

**},**

**getCount: function () {**

**return count;**

**},**

**};**

**})();**

**Counter.increment();**

**Counter.increment();**

**console.log(Counter.getCount());**

**A. 1  
B. 2  
C. undefined  
D. Error  
Answer: B**

**17. Advanced this and bind**

1. **What will the following code output?**

**javascript**

**CopyEdit**

**const person = {**

**name: "Alice",**

**getName: function () {**

**return this.name;**

**},**

**};**

**const getName = person.getName.bind({ name: "Bob" });**

**console.log(getName());**

**A. Alice  
B. Bob  
C. undefined  
D. Error  
Answer: B**

**18. Generators**

1. **What is the purpose of a generator function?  
   A. To generate objects  
   B. To yield multiple values over time  
   C. To replace regular functions  
   D. None of the above  
   Answer: B**
2. **What will this generator output?**

**javascript**

**CopyEdit**

**function\* generator() {**

**yield 1;**

**yield 2;**

**return 3;**

**}**

**const gen = generator();**

**console.log(gen.next());**

**console.log(gen.next());**

**console.log(gen.next());**

**A. { value: 1, done: false } { value: 2, done: false } { value: 3, done: true }  
B. 1, 2, 3  
C. Error  
D. None of the above**