Practical JavaScript Questions (Beginner to Advanced) with Solutions

---

Section 1: Basics & Fundamentals

Q1. Print numbers from 1 to 10 using a loop.

Solution:

```
for(let i = 1; i <= 10; i++) {
    console.log(i);
}
```

Explanation: Using a `for` loop, `i` starts at 1, increments until 10, printing each number.

Q2. Check if a number is even or odd.

Solution:

```
let num = 7;
if(num % 2 === 0) {
    console.log('Even');
} else {
    console.log('Odd');
}
```

Explanation: `%` gives the remainder; if `num % 2 === 0`, it's even.

Q3. Find the factorial of a number.

Solution:

```
function factorial(n) {
    if(n === 0 || n === 1) return 1;
    return n * factorial(n-1);
}
console.log(factorial(5));
```

Explanation: Recursively multiply numbers down to 1.

Q4. Print the multiplication table of a given number.

Solution:

```
let num = 5;
for(let i=1; i<=10; i++) {
    console.log(`${num} x ${i} = ${num*i}`);
}
```

Explanation: Loop multiplies `num` with numbers 1–10.

Q5. Find the sum of first n natural numbers.

Solution:

```
let n = 10;
let sum = (n*(n+1))/2;
console.log(sum);
```

Explanation: Uses the formula for sum of first n numbers.

Section 2: Arrays & Strings

Q6. Reverse an array.

Solution:

```
let arr = [1,2,3,4,5];
let reversed = arr.reverse();
console.log(reversed);
```

Explanation: `reverse()` reverses the array in place.

Q7. Find the largest number in an array.

Solution:

```
let arr = [3,7,1,9,5];
let max = Math.max(...arr);
console.log(max);
```

Explanation: Spread operator `...` passes array elements to `Math.max`.

Q8. Count vowels in a string.

Solution:

```
let str = 'JavaScript';
let count = (str.match(/[aeiou]/gi) || []).length;
console.log(count);
```

Explanation: Regex matches vowels globally and case-insensitively.

Q9. Merge two arrays.

Solution:

```
let arr1 = [1,2,3];
let arr2 = [4,5,6];
let merged = arr1.concat(arr2);
console.log(merged);
```

Explanation: `concat()` joins two arrays.

Q10. Remove duplicates from an array.

Solution:

```
let arr = [1,2,3,2,4,1];
let unique = [...new Set(arr)];
console.log(unique);
```

Explanation: `Set` stores unique values.

Q11. Check if a string is a palindrome.

Solution:

```
let str = 'level';
let reversed = str.split('').reverse().join('');
console.log(str === reversed);
```

Explanation: Reverses string and compares to original.

Q12. Find the index of the largest number in an array.

Solution:

```
let arr = [3,7,2,9,5];
let index = arr.indexOf(Math.max(...arr));
console.log(index);
```

Explanation: Finds max value then gets its index.

Section 3: Functions & Recursion

Q13. Fibonacci series up to n terms.

Solution:

```
function fibonacci(n) {
    let a = 0, b = 1;
    for(let i = 1; i <= n; i++) {
        console.log(a);
        [a, b] = [b, a + b];
    }
}
fibonacci(7);
```

Explanation: Swap and add previous two numbers for the sequence.

Q14. Sum of all elements in an array.

Solution:

```
let arr = [1,2,3,4,5];
let sum = arr.reduce((acc, curr) => acc + curr, 0);
console.log(sum);
```

Explanation: `reduce` accumulates sum of all elements.

Q15. Recursive sum of first n natural numbers.

Solution:

```
function sumN(n) {
    if(n === 0) return 0;
```

```
    return n + sumN(n-1);
}
console.log(sumN(5));
```

Explanation: Recursion adds numbers from n down to 0.

Q16. Check if a number is prime.

Solution:

```
function isPrime(num) {
    if(num <= 1) return false;
    for(let i=2; i<=Math.sqrt(num); i++) {
        if(num % i === 0) return false;
    }
    return true;
}
console.log(isPrime(7));
```

Explanation: Checks divisibility up to sqrt of number.

Q17. Generate array of n random numbers.

Solution:

```
let n = 5;
let arr = Array.from({length: n}, () => Math.floor(Math.random()*100));
console.log(arr);
```

Explanation: `Array.from` creates array with random numbers.

Section 4: DOM & Events

Q18. Change text of a paragraph on button click.

Solution:

```
<p id='para'>Old Text</p>
<button onclick='changeText()'>Click Me</button>
<script>
function changeText() {
    document.getElementById('para').innerText = 'New Text';
```

```
}
</script>
```

Explanation: `onclick` triggers a function that modifies `innerText`.

Q19. Add a new list item dynamically.

Solution:

```
<ul id='list'></ul>
<button onclick='addItem()'>Add Item</button>
<script>
function addItem() {
    let li = document.createElement('li');
    li.textContent = 'New Item';
    document.getElementById('list').appendChild(li);
}
</script>
```

Explanation: Creates `<li>` element and appends to `<ul>`.

Q20. Change background color on hover.

Solution:

```
<div id='box' style='width:100px;height:100px;background:red;'></div>
<script>
let box = document.getElementById('box');
box.addEventListener('mouseover', () => box.style.background = 'green');
box.addEventListener('mouseout', () => box.style.background = 'red');
</script>
```

Explanation: Uses `mouseover` and `mouseout` events.

Section 5: Advanced & Tricky Challenges

Q21. Closure example: counter function.

Solution:

```
function counter() {
    let count = 0;
```

```
    return function() {
        count++;
        return count;
    }
}
let increment = counter();
console.log(increment()); //1
console.log(increment()); //2
```

Explanation: Inner function remembers `count` due to closure.

Q22. Debounce function to limit function calls.

Solution:

```
function debounce(func, delay) {
    let timer;
    return function() {
        clearTimeout(timer);
        timer = setTimeout(() => func.apply(this, arguments), delay);
    };
}
```

Explanation: Ensures function executes after delay, ignoring rapid repeated calls.

Q23. Throttle function example.

Solution:

```
function throttle(func, limit) {
    let lastFunc;
    let lastRan;
    return function() {
        const context = this;
        const args = arguments;
        if(!lastRan) {
            func.apply(context, args);
            lastRan = Date.now();
        } else {
            clearTimeout(lastFunc);
            lastFunc = setTimeout(function() {
                if(Date.now() - lastRan >= limit) {
                    func.apply(context, args);
                    lastRan = Date.now();
                }
```

```
            }, limit - (Date.now() - lastRan));
        }
    }
}
```

Explanation: Limits function execution to once per specified interval.

Q24. Find duplicates in an array.

Solution:

```
let arr = [1,2,3,2,4,5,1];
let duplicates = arr.filter((item, index) => arr.indexOf(item) !== index);
console.log([...new Set(duplicates)]);
```

Explanation: Filters repeated items and removes duplicates using `Set`.

Note: This document can be further extended to include **50–60 questions** covering arrays, strings, recursion, DOM manipulation, events, closures, promises, and other tricky JS scenarios. Copy this content to Word/Google Docs and export as PDF for a complete printable resource.