JavaScript Q&A: Data Types (Numbers, Strings, Boolean, Null, Undefined, BigInt, Symbol)

Q1 [Interview] What are the 7 primitive data types in JavaScript?

Ans: Number, String, Boolean, Null, Undefined, BigInt, Symbol.

Q2 [Interview] What is the type of NaN in JavaScript?

Ans: NaN is of type 'number'.

```
console.log(typeof NaN); // 'number'
```

Q3 [Interview] What is the difference between null and undefined?

Ans: `null` is assigned intentionally; `undefined` means not assigned.

Q4 [Interview] What is BigInt used for?

Ans: To represent numbers larger than Number.MAX_SAFE_INTEGER.

```
let big = 1234567890123456789012345678901234567890n;
```

Q5 [Interview] What is the output of typeof null?

Ans: It's 'object', due to a legacy bug in JavaScript.

```
console.log(typeof null); // 'object'
```

Q6 [Interview] How are strings declared in JavaScript?

Ans: Using '', "", or template literals ``.

```
let s = 'Hello';
let s2 = `Hi ${name}`;
```

Q7 [Interview] Is JavaScript strongly typed?

Ans: No. JavaScript is loosely typed and uses type coercion.

Q8 [Interview] How to check if a variable is a BigInt?

Ans: Use typeof or check for 'n' suffix.

```
typeof 123n // 'bigint'
```

Q9 [Interview] What is a Symbol?

Ans: A unique and immutable primitive value often used as object keys.

```
let id = Symbol('id');
```

Q10 [Interview] Difference between == and ===?

Ans: `==` allows coercion; `===` checks both value and type.

Q11 [Interview] How to find the type of a variable?

Ans: Use typeof operator.

```
typeof 'Hello' // 'string'
```

Q12 [Interview] What are template literals?

Ans: Strings allowing embedded expressions using backticks.

```
`Hello, ${name}`
```

Q13 [Interview] Can BigInt be used with Math methods?

Ans: No. You must convert them to number first.

Q14 [Interview] How to convert string to number?

Ans: Using Number(), parseInt(), or unary +.

```
let x = +'42';
```

Q15 [Interview] What is Boolean conversion of an empty string?

Ans: false

```
Boolean('') // false
```

Q16 [DSA] Count number of numeric values in an array.

Ans: Use typeof inside a loop.

```
const arr = [1, '2', 3n, null, undefined];
let count = 0;
for (let v of arr) {
  if (typeof v === 'number') count++;
}
```

Q17 [DSA] Convert all string numbers in array to actual numbers.

Ans: Use map and typeof check.

```
const arr = ['1', 2, '3'];
let converted = arr.map(v => typeof v === 'string' ? Number(v) : v);
```

Q18 [DSA] Calculate total length of string elements in array.

Ans: Filter strings and use reduce.

```
const arr = ['hello', 123, 'js'];
let total = arr.filter(v => typeof v === 'string').reduce((acc, val) => acc + val.length, 0);
```

Q19 [DSA] Check if array has both null and undefined.

Ans: Use includes method.

```
arr.includes(null) && arr.includes(undefined)
```

Q20 [DSA] Write function to detect data types in an array.

Ans: Return array of types.

```
const detectTypes = arr => arr.map(v => typeof v);
```

Q21 [DSA] Create array from object values that are strings.

Ans: Use Object.values + filter.

```
let obj = {a: 'x', b: 5, c: 'z'};
let res = Object.values(obj).filter(v => typeof v === 'string');
```

Q22 [DSA] Remove non-boolean values from array.

Ans: Use typeof inside filter.

```
arr.filter(v => typeof v === 'boolean');
```

Q23 [DSA] Convert boolean values to 1 (true) or 0 (false).

Ans: Use map and ternary.

```
arr.map(v => typeof v === 'boolean' ? (v ? 1 : 0) : v);
```

Q24 [DSA] Find all unique types in array.

Ans: Use Set and typeof.

```
let unique = [...new Set(arr.map(v => typeof v))];
```

Q25 [DSA] Check if all elements are defined.

Ans: Ensure none are undefined.

```
arr.every(v => v !== undefined);
```

Q26 [DSA] Sum BigInts in an array.

Ans: Use reduce with BigInt addition.

```
arr.reduce((sum, v) => sum + (typeof v === 'bigint' ? v : 0n), 0n);
```

Q27 [DSA] Group values by type.

Ans: Use object with keys as typeof.

```
let grouped = {};
arr.forEach(v => {
  let t = typeof v;
  grouped[t] = (grouped[t] || []).concat(v);
});
```

Q28 [DSA] Find longest string in array.

Ans: Use reduce to compare lengths.

```
arr.reduce((a, b) => a.length > b.length ? a : b);
```

Q29 [DSA] Remove falsy values from array.

Ans: Use Boolean to filter.

```
arr.filter(Boolean);
```

Q30 [DSA] Replace all null/undefined with default value.

Ans: Use map.

```
arr.map(v => v ?? 'default');
```

Q31 [LeetCode] typeof [] + typeof null = ?

Ans: typeof [] = 'object', typeof null = 'object' -> 'objectobject'

Q32 [LeetCode] What is 1 + '2' + 3?

Ans: '12' + 3 = '123'

Q33 [LeetCode] '5' - true = ?

Ans: 5 - 1 = 4 (coercion)

Q34 [LeetCode] '5' + true = ?

Ans: '5true' (string concatenation)

Q35 [LeetCode] '5' * '2' = ?

Ans: 10 (both coerced to numbers)

Q36 [LeetCode] typeof Symbol() === 'symbol'?

Ans: true

Q37 [LeetCode] undefined == null?

Ans: true (loose equality)

Q38 [LeetCode] undefined === null?

Ans: false (strict equality)

Q39 [LeetCode] 'hello' instanceof String?

Ans: false (string literal, not object)

Q40 [LeetCode] typeof NaN === 'number'?

Ans: true

Q41 [Conceptual] Difference between null and undefined?

Ans: null is intentional absence; undefined is default.

Q42 [Conceptual] How to check if a variable is undefined?

Ans: Use === operator.

```
if (value === undefined)
```

Q43 [Conceptual] Does typeof always return string?

Ans: Yes. Always returns string.

Q44 [Conceptual] Is typeof NaN 'number'?

Ans: Yes. By spec.

Q45 [Conceptual] Is typeof function 'object'?

Ans: No. It's 'function'.

Q46 [Conceptual] Can typeof return 'null'?

Ans: No. It returns 'object' for null.

Q47 [Conceptual] Is BigInt used in JSON?

Ans: No. JSON doesn't support BigInt.

Q48 [Conceptual] Can Symbol keys be iterated?

Ans: No. Symbols are non-enumerable.

Q49 [Conceptual] What is the default type of variables?

Ans: undefined if uninitialized.

Q50 [Conceptual] Does JS auto-convert types?

Ans: Yes. Type coercion is common.