JavaScript 50 Q&A: Implicit & Explicit Type Coercion

Q1 [Interview] What is type coercion in JavaScript?

Ans: Automatic or manual conversion between types.

Q2 [Interview] Difference between implicit and explicit coercion?

Ans: Implicit is automatic, explicit is manual using functions.

Q3 [Interview] What does String(123) do?

Ans: Explicitly converts number to string.

```
String(123) // '123'
```

Q4 [Interview] Is +'5' implicit or explicit?

Ans: Explicit (unary plus).

```
+'5' // 5
```

Q5 [Interview] What is result of '5' - 2?

Ans: Implicit coercion, string to number.

```
'5' - 2 // 3
```

Q6 [DSA] Convert array of numbers to strings.

Ans: Use map with String().

```
arr.map(String)
```

Q7 [DSA] Check if coerced value equals original.

Ans: Compare with === after coercion.

Q8 [DSA] Convert string to boolean explicitly.

Ans: Use Boolean().

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

Q9 [DSA] Ensure numeric input stays number.

Ans: Use Number().

```
Number('42') // 42
```

Q10 [DSA] Convert all falsy values to 0.

Ans: Use map and Boolean().

```
arr.map(v => Boolean(v) ? v : 0)
```

Q11 [LeetCode] typeof +true?

Ans: 'number' (true coerced to 1)

Q12 [LeetCode] '5' * null?

Ans: 0 (string becomes 5, null becomes 0)

Q13 [LeetCode] [] + []?

Ans: '' (both become empty strings)

Q14 [LeetCode] [] + {}?

Ans: '[object Object]'

Q15 [LeetCode] {} + []?

Ans: 0 or '[object Object]' depending on parsing context

Q16 [Conceptual] Does JS coerce in '==' comparisons?

Ans: Yes, '==' allows type coercion.

Q17 [Conceptual] Is NaN coerced to number?

Ans: Yes, NaN is of type number.

Q18 [Conceptual] Which is safer: == or ===?

Ans: === avoids coercion.

Q19 [Conceptual] Is Boolean('false') true?

Ans: Yes. All non-empty strings are truthy.

Q20 [Conceptual] What happens if '5' + 1?

Ans: String concatenation due to '5'.