```
console.clear();

/*

======================================================================

                Data Types Interview Questions 🔥

======================================================================

*/

console.log("Data Types Interview Questions");

/*

1. What is the difference between null and undefined in JavaScript?

👉 `null` is an assigned value that represents the intentional absence of any object value.

👉 `undefined` means a variable is declared but not assigned any value.

*/

// Example:

let a = null;

let b;

console.log("a:", a, "Type:", typeof a); // object (JS bug)

console.log("b:", b, "Type:", typeof b); // undefined

/*

2. What is the purpose of the `typeof` operator in JavaScript?
```

```
38
39  👉 The `typeof` operator is used to find the data type of a variable.
40
41  */
42
43  // Example:
44
45  console.log(typeof "JavaScript"); // string
46
47  console.log(typeof 42); // number
48
49  console.log(typeof false); // boolean
50
51  /*
52
53  3. What are the primitive data types in JavaScript?
54
55  👉 JavaScript has 7 primitive data types:
56
57     ✅ Number
58
59     ✅ String
60
61     ✅ Boolean
62
63     ✅ Undefined
64
65     ✅ Null
66
67     ✅ BigInt
68
69     ✅ Symbol
70
71  */
72
73  // Example:
74
75  let myFavoriteNumber = -7; // number
```

```
76
77  let myFavoriteString = "JavaScript"; // string
78
79  let myFavoriteBoolean = true; // boolean
80
81  let myFavoriteUndefined; // undefined
82
83  let myFavoriteNull = null; // null
84
85  let myFavoriteBigInt = 12345678901234567890123456789012345678901234567890n; // bigint
86
87  let myFavoriteSymbol = Symbol("id"); // symbol
88
89  /*
90
91  4. What is the difference between implicit and explicit type coercion?
92
93  👉 Implicit Coercion happens automatically by JavaScript.
94
95  👉 Explicit Coercion happens when we manually convert a value.
96
97  */
98
99  // Implicit
100
101 console.log(10 + "5"); // "105"
102
103 // Explicit
104
105 console.log(Number("5") + 10); // 15
106
107 /*
108
109 5. Convert a number to a string.
110
111 👉 Use `toString()` or concatenate with `""`.
112
113 */
```

```
114
115 let num = 123;
116
117 console.log(num.toString(), typeof num.toString()); // "123" string
118
119 console.log("" + num, typeof ("" + num)); // "123" string
120
121 /*
122
123 6. Convert a string to a number.
124
125 👉 Use `Number()` or `+` operator.
126
127 */
128
129 console.log(Number("50"), typeof Number("50")); // 50 number
130
131 console.log(+"50", typeof +"50"); // 50 number
132
133 /*
134
135 7. What is the difference between parseInt() and parseFloat()?
136
137 👉 `parseInt()` converts a string to an integer.
138
139 👉 `parseFloat()` converts a string to a decimal.
140
141 */
142
143 console.log(parseInt("99.99")); // 99
144
145 console.log(parseFloat("99.99")); // 99.99
146
147 /*
148
149 8. What are truthy and falsy values?
150
151 👉 Truthy values: non-empty strings, numbers (except 0), true, objects, arrays.
```

```
152
153 👉 Falsy values: false, 0, "", null, undefined, NaN.
154
155 */
156
157 console.log(Boolean("hello")); // true
158
159 console.log(Boolean("")); // false
160
161 /*
162
163 9. How to check if a variable is `NaN`?
164
165 👉 Use `Number.isNaN(value)`
166
167 */
168
169 console.log(Number.isNaN(NaN)); // true
170
171 console.log(Number.isNaN("hello")); // false
172
173 /*
174
175 10. What is the output of `typeof NaN`?
176
177 👉 "number" — NaN is still considered a numeric value in JavaScript.
178
179 */
180
181 console.log(typeof NaN); // number ❗(weird JS behavior)
182
183 /*
184
185 11. How to check if a number is finite?
186
187 👉 Use `Number.isFinite(value)`
188
189 */
```

```
console.log(Number.isFinite(100)); // true

console.log(Number.isFinite(Infinity)); // false

/*

12. What is Symbol in JavaScript?

👉 `Symbol` is a unique primitive value used for object property keys to avoid name collisions.

*/

const sym1 = Symbol("id");

const sym2 = Symbol("id");

console.log(sym1 === sym2); // false

/*

13. What is BigInt in JavaScript?

👉 `BigInt` is a primitive used to represent very large integers beyond Number.MAX_SAFE_INTEGER.

*/

let bigNum = 9876543219876543219876543219n;

console.log(bigNum, typeof bigNum); // bigint

/*

14. What is the default value of uninitialized variables?

👉 If a variable is declared but not assigned, it holds the value `undefined`.

*/
```

```
228
229 let x;
230
231 console.log(x); // undefined
232
233 /*
234
235 15. Can typeof null be "object"?
236
237 👉 Yes, due to a historical bug in JavaScript, `typeof null` returns `"object"`.
238
239 */
240
241 console.log(typeof null); // object ❗
242
243 /*
244
245 ========================================================================
246
247          Output-Based JavaScript Interview Questions 🔥
248
249 ========================================================================
250 */
251
252 console.log("Output-Based Interview Questions");
253
254 // 1️⃣ String + Number
255
256 console.log("10" + 20); // "1020"
257
258 // 2️⃣ Number - String
259
260 console.log(10 - "5"); // 5 (string converted to number)
261
262 // 3️⃣ Boolean + Number
263
264 console.log(true + 1); // 2 (true is 1)
265
```

```
// 4 Boolean - Boolean

console.log(false - true); // -1 (false = 0, true = 1)

// 5 Null + Number

console.log(null + 10); // 10 (null treated as 0)

// 6 Undefined + Number

console.log(undefined + 10); // NaN (undefined cannot be converted)

// 7 Empty String + Number

console.log("" + 10); // "10" (string concatenation)

// 8 Empty String - Number

console.log("" - 10); // -10 ("" is treated as 0)

// 9 Comparing null and 0

console.log(null == 0); // false

console.log(null >= 0); // true ❗ (unexpected behavior)

// 10 Comparing undefined and null

console.log(undefined == null); // true

console.log(undefined === null); // false

// 11 Comparing Boolean values

console.log(true == "1"); // true (string "1" is converted to number)
```