



LEARN JAVASCRIPT



**Enhance Your JavaScript Skills:
Explore Variables and Data Types
Through Interactive Questions!**



Coding Questions with answers and explanations!

Question: How do you declare a variable in JavaScript and assign the string "Hello, World" to it?

Answer:

```
let message = "Hello, World";
```

Explanation: The let keyword is used to declare a variable named message, and the = operator is used to assign the string "Hello, World" to it. let allows the variable to be reassigned later if needed.

Question: What will be the output of the following code?

```
const a = 5;  
const b = '5';  
console.log(a + b);
```

Answer: The output will be "55".

Explanation: In this case, JavaScript performs type coercion. The number 5 (from variable a) is converted to a string and concatenated with the string '5' (from variable b), resulting in the string "55".

Question: Identify the data types of the following variables:

let a = true;

let b = 1;

let c = "Hello";

let d = null;

let e;

Answer:

a: Boolean

b: Number

c: String

d: Object (specifically null)

e: Undefined

Explanation:

true is a boolean value.

1 is a number.

"Hello" is a string.

null is an object, a special keyword denoting a null value.

By default, variables are undefined if no value has been assigned.

Question: What is the difference between using let and const for declaring variables?

Answer: Variables declared with let can be reassigned, while variables declared with const cannot.

Explanation: let is used for variables whose value may change, while const is used for variables that should remain constant throughout the script.

Question: How does JavaScript handle adding a number and a string together?

Answer: JavaScript converts the number to a string and then concatenates the two strings.

Explanation: This process is known as type coercion. If you add a number (like 5) and a string (like "10"), JavaScript treats them both as strings and performs string concatenation, resulting in "510".

Question: What will be the output of the following code snippet?

```
let x = "5";  
let y = 3;  
console.log(typeof(x * y));
```

Answer: The output will be "number".

Explanation: JavaScript converts the string "5" to a number in order to perform the multiplication. The result of the operation $x * y$ is a number, so `typeof(x * y)` returns "number".

Question: How can you convert the string "123" into a number in JavaScript?

Answer:

```
let num = parseInt("123");
```

Explanation: The `parseInt()` function parses a string argument and returns an integer of the specified radix (base in mathematical numeral systems). In this case, it converts the string "123" into the number 123.

Question: What will be the output of the following code?

```
var a;
```

```
console.log(typeof a);
```

Answer: The output will be "undefined".

Explanation: The variable `a` is declared but not initialized, so its value is undefined. When using `typeof` on an uninitialized variable, it returns the string "undefined".

Question: What will be the output of the following code and why?

```
let x = null;  
console.log(typeof x);
```

Answer: The output will be "object".

Explanation: Although null is supposed to represent the absence of a value, in JavaScript, it is considered as an object. This is a well-known quirk of JavaScript, where `typeof null` returns "object".

Question: Can you reassign a value to a variable declared with `const`? What will happen if you try to do so?

Answer: No, you cannot reassign a value to a variable declared with `const`. An error will be thrown if you try to do so.

Explanation:

```
const a = 5;  
a = 10; // This will throw an error.
```

When you declare a variable with `const`, its value is meant to be constant and cannot be changed throughout the script. Attempting to reassign a value to a `const` variable results in a `TypeError`.