

# JavaScript Quiz Questions and Answers



JavaScript  
Code Examples and Questions



Laurence Svekis BaseScripts.com

|  |   |
|--|---|
| What is a closure in JavaScript and how do you create one?       | 1 |
| What is the difference between == and === in JavaScript?         | 2 |
| How do you declare a variable in JavaScript?                     | 2 |
| What is the difference between null and undefined in JavaScript? | 3 |
| How do you check if a variable is an array in JavaScript?        | 3 |
| What is the difference between var and let in JavaScript?        | 4 |
| How do you create a new object in JavaScript?                    | 5 |
| How do you add an element to the end of an array in JavaScript?  | 5 |
| How do you loop through an object in JavaScript?                 | 6 |
| How do you check if a variable is defined in JavaScript?         | 6 |

## What is a closure in JavaScript and how do you create one?

A closure is a function that has access to variables in its outer (enclosing) function's scope chain. This allows the closure to retain access to those variables even after the outer function has returned. Closures are created by defining a function inside another function and returning it. Here is an example:

```
function outerFunction() {
```

```
let outerVariable = "Hello";

function innerFunction() {
  console.log(outerVariable);
}

return innerFunction;
}

let closure = outerFunction(); // closure now refers to
innerFunction
closure(); // logs "Hello" because innerFunction has
access to outerVariable
```

## What is the difference between == and === in JavaScript?

The == operator compares values for equality after performing type coercion, while the === operator compares values for equality without performing type coercion. For example:

```
1 == "1" // true because "1" is coerced to a number
1 === "1" // false because "1" is not equal to 1
without coercion
```

## How do you declare a variable in JavaScript?

You can declare a variable in JavaScript using the var, let, or const keywords. var and let are used for declaring variables that

can be reassigned, while `const` is used for declaring variables that cannot be reassigned. For example:

```
var x = 5; // var keyword
let y = 10; // let keyword
const z = 15; // const keyword
```

## What is the difference between `null` and `undefined` in JavaScript?

`null` is a value that represents the intentional absence of any object value, while `undefined` is a value that represents an uninitialized or nonexistent value. For example:

```
let x = null; // x is intentionally set to null
let y; // y is undefined because it has not been
initialized
```

## How do you check if a variable is an array in JavaScript?

You can check if a variable is an array in JavaScript using the `Array.isArray()` method. For example:

```
let x = [1, 2, 3];
let y = "hello";
console.log(Array.isArray(x)); // true
console.log(Array.isArray(y)); // false
```

## What is the difference between var and let in JavaScript?

The main difference between var and let in JavaScript is that var has function scope, while let has block scope. This means that variables declared with var are accessible within the entire function in which they are declared, while variables declared with let are only accessible within the block in which they are declared. For example:

```
function example() {  
  var x = 5;  
  if (true) {  
    var x = 10;  
  }  
  console.log(x); // logs 10 because x is reassigned  
  // within the same function scope  
}
```

```
function example2() {  
  let y = 5;  
  if (true) {  
    let y = 10;  
  }  
  console.log(y); // logs 5 because y is only  
  // accessible within the block scope of the if statement  
}
```

## How do you create a new object in JavaScript?

You can create a new object in JavaScript using either object literal notation `{}`, constructor notation `new Object()`, or class notation `class MyClass {}`. For example:

```
let obj1 = {}; // object literal notation
let obj2 = new Object(); // constructor notation
class MyClass { // class notation
  constructor(x, y){
    this.x = x;
    this.y = y;
  }
}

let obj3 = new MyClass(5, 10);
```

## How do you add an element to the end of an array in JavaScript?

You can add an element to the end of an array in JavaScript using the `push()` method. For example:

```
let arr = [1, 2, 3];
arr.push(4);
console.log(arr); // logs [1, 2, 3, 4]
```

## How do you loop through an object in JavaScript?

You can loop through an object in JavaScript using a `for...in` loop. For example:

```
let obj = {x: 1, y: 2, z: 3};
for (let prop in obj) {
  console.log(`${prop}: ${obj[prop]}`);
}
// logs "x: 1", "y: 2", "z: 3"
```

## How do you check if a variable is defined in JavaScript?

You can check if a variable is defined in JavaScript using the `typeof` operator or the `undefined` keyword. For example:

```
let x;
console.log(typeof x === "undefined"); // true because
x has not been initialized
console.log(x === undefined); // true because x has not
been initialized
```

