

Slay the JavaScript Interview

100 answers that every developer needs to know

JSTips

Contents

Hi there!	2
What are closures in JavaScript?	3
What is the difference between slice and splice? . . .	4
What is scope in JavaScript?	5
Why JavaScript is a dynamic and weakly type language?	6
When function definition is not hoisted in JavaScript?	7
What is a freeze method?	8

Hi there!

Thanks for download the sample. If you like it or hated, you can shoot your thoughts in this **short survey** and receive an extra 15% off. I want to make this book usable and valuable for everybody, so I'll appreciate your feedback!

Thanks so much! Enjoy!

What are closures in JavaScript?

Closures are mechanisms used for enabling data privacy by combining a function enclosed with reference to its surrounding environment. A closure function has access to the parent scope even after the parent function has closed.

```
/*
 * Example of simple closure
 */
function parentFunction() {
    let outerVar = 'I am in the parent scope!';
    function innerFunction() {
        // Variable outerVar is store in the scope
        console.log(outerVar);
    }
    return innerFunction;
}
```

```
const myClosure = parentFunction();
myClosure();
// Output: I am in the parent scope!
```

```
/*
 * Example of usefull closure
 */
function multiply(a) {
    return function innerMultiply(b) {
        // Variable a is store in the scope
        return a * b;
    }
}
```

```
const double = multiply(2);
double(3);
// Output: 6
double(8);
// Output: 16
```

What is the difference between slice and splice?

Use `slice` to pick elements from an array that you do not want to mutate. Consider use `splice` when you want to modify the original array inserting and removing elements.

```
const food = ['burrito', 'pasta', 'noodles'];
```

```
// Using slice to pick elements form an array based on elements index
```

```
const myFood = food.slice(1, 2);
```

```
// myFood is ['pasta']
```

```
// food is ['burrito', 'pasta', 'noodles']
```

```
// Using splice to update the orignal array based on elements index
```

```
food.splice(1, 1, 'burger');
```

```
// Output: ['pasta']
```

```
// food is ['burrito', 'burger', 'noodles']
```

What is scope in JavaScript?

The scope of a variable refers to the lifetime and accessibility of variables within the context of code and is generally of two types, global and local. A global variable is accessible outside the block where it is declared, whereas local variables will only be accessible within the block where it is declared.

```
// Initialize a global variable
var globalVariable = "I'm a global variable";

const testingScopes = () => {
  // Initialize a local variable only visible in this block
  var localVariable = "I'm a local variable";
  // Log global variable as functions has access to the outer scope
  console.log(globalVariable);
};

// Log global and local variables
console.log(globalVariable);
// Output: "I'm a global variable"
console.log(localVariable);
// Output: ReferenceError: localVariable is not defined
testingScopes();
// Output: "I'm a global variable"
```

Why JavaScript is a dynamic and weakly type language?

JavaScript is dynamic because variables can be created at the runtime, and the type of them will be determined at code execution. Moreover, JavaScript is a weakly typed dynamic language since it does not need to specify the type of variable that will be stored. The language automatically assigns a type to the variable during runtime.

```
var thisIsAString = 'I am a string';  
> typeof thisIsAString;  
// Output: string
```

```
// Assign an integer  
thisIsAString = 1;  
> typeof thisIsAString;  
// Output: number
```

When function definition is not hoisted in JavaScript?

In JavaScript, unlike variable declaration, function declaration hoists actual function definition. Thus, functions are not hoisted if you assign the function to a variable.

```
console.log(foo());
```

```
// Output: Hello!
```

```
console.log(bar());
```

```
// Output: Uncaught TypeError: bar is not a function
```

```
// Entire function hoisted
```

```
function foo() {  
    return 'Hello!';  
};
```

```
// Only definition hoisted
```

```
var bar = function () {  
    return 'Bye!';  
};
```

```
console.log(bar());
```

```
// Output: return 'Bye!'
```


What is a freeze method?

The `freeze()` method freezes the modification of existing property, value, and attribute. In addition to that, the `freeze()` method also prevents its prototype from being modified.

```
const user = {};  
user.name = 'Mikey';  
const myUser = Object.freeze(user.name);
```

```
Object.isFrozen(myUser); //true  
user === myUser // true
```

```
user.age = '21';  
// Output: TypeError: Cannot add property name, object is not extensible
```