**JAVASCRIPT INTERVIEW QUESTIONS**

**1 – What are the JavaScript data types?**

**Answer:**

1. Number
2. String
3. Boolean
4. Object
5. Symbol
6. Undefined
7. Null

**2 – What are the differences between global and local variables? And how can having too many global variables negatively affect your code?**

**Answer:**

Global variables have no scope and are accessible globally — they’re fully available and accessible throughout your code.

However, global variables can also be more difficult to debug as coupling occurs between the many global variables, making bugs more likely, as well as more difficult to find and debug. There can also be conflicts between variable names in the local and global scope.

## 3 – What are the differences between the =, ==, and === operators?

**Answer:**

The = operator is for saying a variable is equal to something when declaring it, such as a string or a number:

The == abstract equality operator checks for equality only in value, and is a less strict evaluator than the === operator. The == operator converts the variable values to the same type before comparing them — also known as type coercion.

Therefore, even if two different variables are named, such as the number 2 and string'2', as variables are converted to the same value before comparing them, this will still evaluate to true:

The === strict equality operator does not use any form of type coercion, and to evaluate as true both the type and the value must be the same:

## 5 – Explain the .pop() and .push() methods using the following array:

let bootdotdev = ["the", "best", "coding", "courses", "ever", "like", "totally"];

1. What would happen if you used bootdotdev.pop()
2. What would happen if you used bootdotdev.push("definitely", "start", "it")

**Answers:**

The .pop() method removes the last element in the array and returns it. In the array above, bootdotdev.pop() would return "totally".

The .push() method instead adds any included elements to the end of the array, and returns the new length of that array. In the array above, pushing:

bootdotdev.push("definitely", "start", "it")

would return the length 10 , and if you console.log(bootdotdev) you would get:

["the","best","coding","courses","ever","like","totally","definitely","start","it"]

## 6 – What are higher order functions, and what are their benefits?

**Answer:**

Higher-order functions are functions that either accept or return another function — by taking them as an argument, or by returning them. They allow for common patterns that support the .map, .sort or .filter functions, for example.

Take a look at how the map function accepts a function as a parameter and applies it to the given array.

## 7 – Write a loop that prints every number divisible by 3 from 1-200:

**Answer:**

for (let i = 1; i < 201; i++) {

if (i % 3 === 0) {

console.log(i)

}

}

## 8 – Name all JavaScript’s Boolean logical operators:

**Answer:**

The AND && operator, the NOT ! operator, and the OR || operator.

The && AND operator requires both operands to be true to evaluate to true. For example:

The || OR operator requires one of the operands to be true to evaluate to true. For example:

The ! NOT operator evaluates to true if the two operands are not equal. For example:

## ****10 – What is JSON? And how would you convert JSON strings into objects, or convert an object into a JSON string?****

**Answer:**

JSON, or JavaScript Object Notation, is used to send and convert plain text into JavaScript objects, and they are written in exactly the same way. JSON is often used to display data on a web page after being read from a web server.

It’s used to send data between computers, but can also be used by any programming language — it isn’t limited to JavaScript. The text-only format makes it easy to send between computers.

To convert a JSON string into a JavaScript object, use this function: JSON.parse()

To convert an object into a JSON string, use this function: JSON.stringfy()

## 11 – What are the .slice() and .splice() methods, and how do they differently affect arrays?

**Answer:**

.slice() doesn’t modify the original array, and returns the elements from the array, minus the instructed elements to be sliced off based on the numbers instructed in the brackets.

If one parameter is written, this will be used as the start parameter, and if two parameters are written, the second element will be used as the end parameter.

The .splice() method instead modifies the original array. It returns the deleted elements as arrays, and is often use to insert or remove elements to or from an existing array.

## 12 – What are arrow functions and what benefits do they bring in your code?

**Answer:**

Arrow functions were introduced in ES6 and are a shorthand version of writing traditional functions. They save room and can make code more easily readable, are quicker to write, and can make coding more efficient. They also inherit the parent version of this.

## 13 – What is the isNaN function?

**Answer:**

The isNaN function determines whether a value is, or is not, a number (Not-a-Number). If the value is not a number, it will evaluate to true, and if the value is a number, it will evaluate to false.

## 14 – What are the differences between undeclared and undefined variables?

**Answers:**

**Undeclared variable:** a variable that isn’t declared as a variable using the var , const or let keywords.

**Undefined variable:** a variable that has been defined using one of the variable keywords, but hasn’t been given a value.

## 15 – What is the DOM? And what does it do?

**Answer:**

The Document Object Model represents HTML documents in a way JavaScript can interpret.

The DOM is an object-oriented view of the HTML web page document where each HTML attribute is an object within a hierachy consisting of element, attribute and body nodes that can then be accessed and modified or manipulated using functions.

### **16. Is JavaScript a case-sensitive language?**

Yes, JavaScript is a **case sensitive** language.  The language keywords, variables, function names, and any other identifiers must always be typed with a consistent capitalization of letters.

### **17. How can you create an object in JavaScript?**

JavaScript supports **Object** concept very well. You can create an object using the **object literal** as follows −

|  |  |
| --- | --- |
|  | var emp = {  name: "Daniel",  age: 23}; |

### **18. Can you assign an anonymous function to a variable and pass it as an argument to another function?**

Yes! An anonymous function can be assigned to a variable. It can also be passed as an argument to another function.

In case you are facing any challenges with these JavaScript Interview Questions, please comment on your problems in the section below.

### **19. What is the purpose of ‘This’ operator in JavaScript?**

The JavaScript **this** keyword refers to the object it belongs to. This has different values depending on where it is used. In a method, this refers to the owner object and in a function, this refers to the global object.

### **20. What is Callback?**

A **callback** is a plain JavaScript function passed to some method as an argument or option. It is a function that is to be **executed** after another function has finished executing, hence the name ‘**call back**‘. In JavaScript, functions are objects. Because of this, functions can take functions as arguments, and can be returned by other functions.

### **21. What is Closure? Give an example.**

**Closures** are created whenever a variable that is defined outside the **current scope** is accessed from within some inner scope. It gives you access to an outer function’s scope from an inner function. In JavaScript, closures are created every time a function is created. To use a closure, simply define a function inside another function and expose it.

### **22. What is the difference between Attributes and Property?**

**Attributes-**  provide more details on an element like id, type, value etc.

**Property-**  is the value assigned to the property like type=”text”, value=’Name’ etc.

### **23. List out the different ways an HTML element can be accessed in a JavaScript code.**

Here are the list of ways an HTML element can be accessed in a Javascript code:  
(i) **getElementById(‘idname’):** Gets an element by its ID name  
(ii) **getElementsByClass(‘classname’):** Gets all the elements that have the given classname.  
(iii) **getElementsByTagName(‘tagname’):** Gets all the elements that have the given tag name.  
(iv) **querySelector():** This function takes css style selector and returns the first selected element.

### **24. What are the ways to define a variable in JavaScript?**

The three possible ways of defining a variable in JavaScript are:

* **Var** – The JavaScript variables statement is used to declare a variable and, optionally, we can initialize the value of that variable. Example: var a =10; Variable declarations are processed before the execution of the code.
* **Const** – The idea of const functions is not allow them to modify the object on which they are called. When a function is declared as const, it can be called on any type of object.
* **Let** – It is a signal that the variable may be reassigned, such as a counter in a loop, or a value swap in an algorithm. It also signals that the variable will be used only in the block it’s defined in.

### **25. What is the difference between Local storage & Session storage?**

**Local Storage** – The data is not sent back to the server for every HTTP request (HTML, images, JavaScript, CSS, etc) – reducing the amount of traffic between client and server. It will stay until it is manually cleared through settings or program.

**Session Storage** – It is similar to local storage; the only difference is while data stored in local storage has no expiration time, data stored in session storage gets cleared when the page session ends. Session Storage will leave when the browser is closed.

### **26. What is the difference between innerHTML & innerText?**

**innerHTML** – It will process an HTML tag if found in a string

**innerText** – It will not process an HTML tag if found in a string

### **27. What would be the result of 2+5+”3″?**

Since 2 and 5 are integers, they will be added numerically. And since 3 is a string, its concatenation will be done. So the result would be 73. The ” ” makes all the difference here and represents 3 as a string and not a number.

### **28. Name some ways to handle asynchronous operation in javascript**

* **Callback** is a function that is used to notify the calling instance
* **Promise** is an object representing the eventual completion or failure of an asynchronous operation. A pending promise can either be fulfilled with a value or rejected with a reason.  
  Callbacks are attached to the returned promises that make handling of asynchronous code easier and more readable.
* **async/await** is a new addition to ES2017 which is syntactic sugar on top of promises and make asynchronous code look synchronous code

### **29. What is the difference between null and undefined?**

**null** type is an object that is explicitly assigned to a variable.

**undefined** type is undefined where the variable has been declared but has no assigned value

### **30. Explain the difference between synchronous and asynchronous.**

* **Synchronous** is blocking operation while **asynchronous** is not. Synchronous complete the current code before the next code is executed while asynchronous continue on the next code without completing the current code

**31.What are the pop-up boxes types available in JavaScript?**

There are 3 pop up boxes: Alert, Confirm and Prompt.

#### **32. What do you understand by Closures in JavaScript?**

Closures provide a better, concise, creative, and expressive writing code for JavaScript developers and programmers. Technically speaking, closures are a combination of lexical environment and function.

In other words, a closure is a locally declared variable that is related to a function and stays in the memory when the related function has returned. The closure contains all local variables that were in-scope at the time of the closure creation.