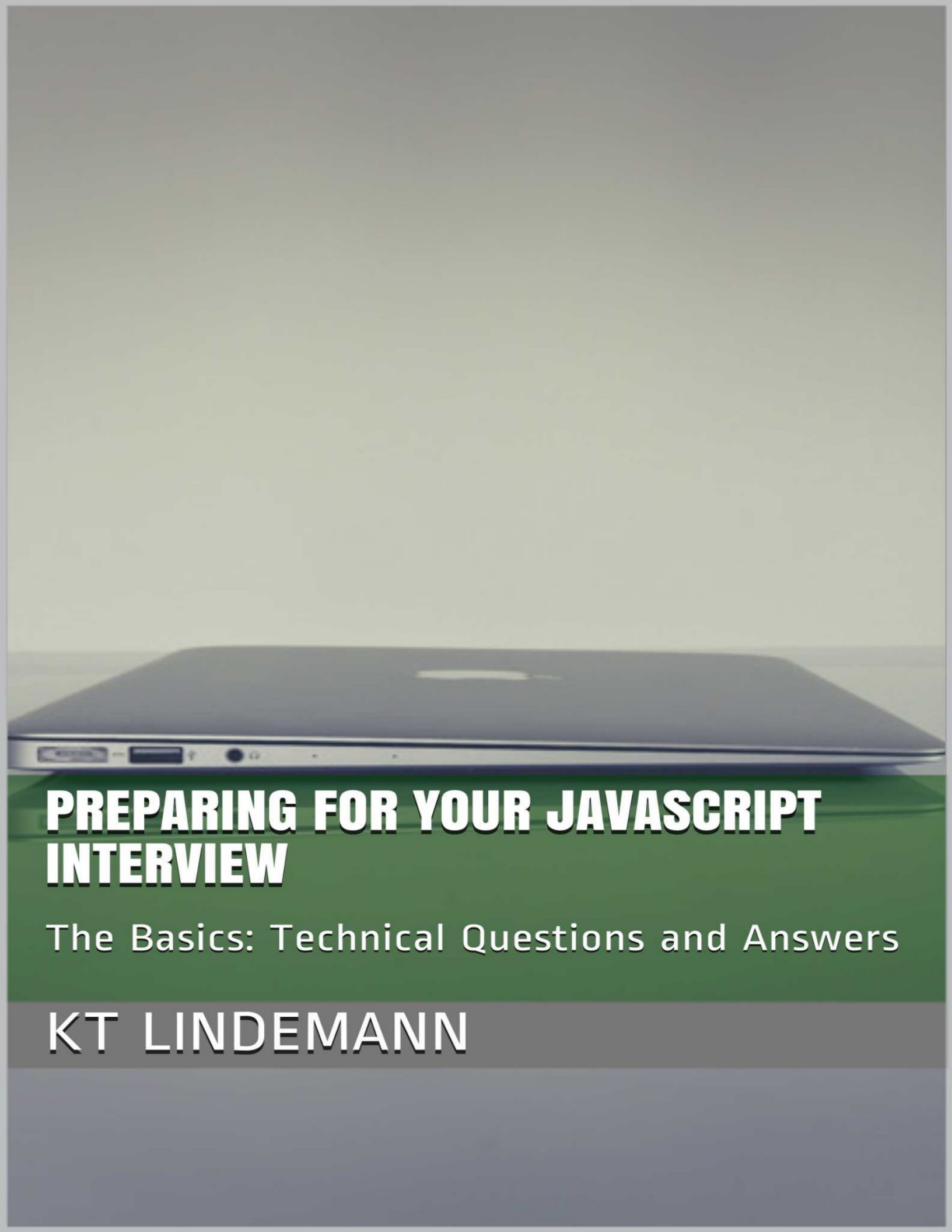

PREPARING FOR YOUR JAVASCRIPT INTERVIEW

The Basics: Technical Questions and Answers

KT LINDEMANN

Preparing for Your JavaScript Interview

The Basics: Technical Questions and Answers

By K.T. Lindemann

Introduction

Today, you cannot be a web developer and not know JavaScript. What started out as a small client side scripting language has evolved into countless frameworks and libraries that are part of every user's web experience. All modern web browsers implement the JavaScript language, thus making it the only one deployed on all personal computers and mobile devices around the world.

As it evolved, JavaScript has embraced advanced features such as prototypal inheritance, modules, namespaces, anonymous (lambda) functions, promises, and even metaprogramming. Thanks to frameworks like NodeJs, you can even use it server-side now.

JavaScript's growth presents a challenge for new developers. On the one hand, the barrier to entry is low – anyone can open a developer tools window in the web browser and begin to write JavaScript. On the other hand, the level of complexity grows exponentially as you begin to explore the functionality available in the vast JavaScript ecosystem.

It's now common for technical phone screens to include JavaScript questions. As a candidate, where should you begin to study?

While other books tend to focus on the coding part of technical interviews, this book aims to run through some common, foundational JavaScript interview questions that candidates may be asked to explain or describe out loud. Focusing on vanilla JavaScript, this book covers the basics of JavaScript data structures, error handling, UI manipulation, and more! Being able to answer these questions concisely and confidently will go a long way towards making you a better candidate.

Good luck!

Table of Contents

General Practice

This section focuses on the basics of vanilla JavaScript.

Question: What is JavaScript?

JavaScript, often abbreviated as "JS", is a high-level, dynamic, untyped, and interpreted run-time language.

JavaScript is a client-side as well as server side scripting language that can be inserted into HTML pages and is understood by web browsers. JavaScript can also be an Object Oriented Programming language

Question: What are the two basic groups of dataypes in JavaScript?

Primitive types: string, number, boolean, null, undefined
Complex types: object (including functions)

Question: What are the JavaScript types?

There are just **six** types in JavaScript: Object, Number, String, Boolean, Null, and Undefined.

Objects include arrays, functions, and ordinary objects.

Numbers may be integers or floating point or the special values NaN and Infinity.

Strings include the empty string.

Question: Which company developed JavaScript?

Netscape is the software company who developed JavaScript.

Question: What are undeclared and undefined variables?

Undeclared variables are those that do not exist in a program and are not declared. If the program tries to read the value of an undeclared variable, then a runtime error is encountered.

Undefined variables are those that are declared in the program but have not been given any value. If the program tries to read the value of an undefined variable, an undefined value is returned.

Question: What are global variables?

Global variables are those that are available throughout the length of the code, that is, these have no scope. The var keyword is used to declare a local variable or object. If the var keyword is omitted, a global variable is declared.

The problems that are faced by using global variables are the clash of variable names of local and global scope. Also, it is difficult to debug and test the code that relies on global variables.

Question: What is ‘this’ keyword in JavaScript?

‘This’ keyword refers to the object from where it was called.

Question: What is the use of isNaN function?

isNan function returns true if the argument is not a number otherwise it is false.

Question: What is negative infinity?

Negative Infinity is a number in JavaScript which can be derived by dividing negative number by zero.

Question: Which symbol is used for comments in Javascript?

// for Single line comments

/* MultiLine
Comment */

Question: What is the === operator?

=== is called as strict equality operator which returns true when the two operands are having the same value without any type conversion.

Question: What is called Variable typing in Javascript?

Variable typing is used to assign a number to a variable and the same variable can be assigned to a string. This is called variable typing.

Question: Does JavaScript support automatic type conversion?

Yes, JavaScript does support automatic type conversion.

JavaScript is a **"loosely typed" language**, which means that whenever an operator or statement is expecting a particular data-type, JavaScript will automatically convert the data to that type

For example, automatic type conversion is often used for checking the existence of DOM elements. Because an Object will automatically convert to a Boolean when used in a conditional statement, this code is possible:

```
var element = document.getElementById("whatever");

if(element) {
  //the element exists
}
else {
  //the element doesn't exist
}
```

Question: What does NULL mean in Javascript?

The NULL value is used to represent no value or no object. A variable that is null indicates that it points to nothing. In APIs, null is often retrieved in a place where an object can be expected but no object is relevant.

Question: What is an undefined value in JavaScript?

Undefined is one of JavaScript's primitive types.

A variable that has not been assigned a value is of type undefined. A method or statement can also be undefined if it tries to use a variable that is undefined. Finally, a function returns undefined if a value was not returned.

Question: What are all the looping structures in JavaScript?

Answer:
In JavaScript we have the following looping statements:

while	loops through a block of code while a condition is true
do...while	loops through a block of code once, and then repeats the loop while a condition is true
for	run statements a specified number of times

Question: What is the break statement?

The **break** statement exits from the current loop.

Question: What is the continue statement?

The **continue** statement continues with next statement of the loop.

Question: Explain how to detect the operating system on the client machine?

The navigator.appVersion string (property) can be used to detect the operating system of the client machine.

Question: What is void(0) used for?

The void operator evaluates the given expression and then returns undefined.

Void(0) is used to prevent the page from refreshing. The parameter “zero” is passed while calling it.

Question: What are the escape characters?

JavaScript uses the \ (backslash) as an escape characters for:

\' single quote
\" double quote
\ backslash
\n new line
\r carriage return
\t tab
\b backspace
\f form feed
\v vertical tab
\0 null character (

Question: Explain how timers work in JavaScript.

A block of JavaScript code is generally executed synchronously. Timers allow us to delay the execution of arbitrary instructions:

setTimeout()
The setTimeout() function is commonly used if you wish to have your function called **once** after the specified delay.

setInterval()
The setInterval() function is commonly used to set a delay for functions that are executed more than once.

setImmediate()
The setImmediate() function can be used instead of the setTimeout(fn, 0) method to execute heavy operations in IE.

Question: What is the use of typeof operator?

'Typeof' is an operator which is used to return a string description of the type of a variable.

Type	Result
Undefined	"undefined"
Null	"object" (see below)
Boolean	"boolean"
Number	"number"
String	"string"
Symbol (new in ECMAScript 2015)	"symbol"
Host object (provided by the JS environment)	*Implementation-dependent*
Function object (implements [[Call]] in ECMA-262 terms)	"function"
Any other object	"object"

Question: Which keywords are used to handle exceptions?

The try/catch/finally statement handles some or all of the errors that may occur in a block of code, while still running code.

The try statement allows you to define a block of code to be tested for errors while it is being executed.

The catch statement allows you to define a block of code to be executed, if an error occurs in the try block.

The finally statement lets you execute code, after try and catch, regardless of the result.

```
try {
   // try to run this code
}
catch(err) {
   // catch errors here
}
finally {
            // run this code regardless of what happens
}
```

Question: How generic objects can be created?

A generic object is essentially an empty blueprint for an object. You can define it anywhere in your Javascript code. An example is below:

```
function Employee(first, last, ssn, salary)
{
   this.firstName = first;
   this.lastName = last;
   this.social = ssn;
   this.pay = salary;
}
```

This creates a generic blueprint for an object known as Employee. To get an instance of Employee, you would call:

```
var john = new Employee("John", "Smith", "050529994", 72534);
```

Question: What is the use of Push method in JavaScript?

The push method is used to add or append one or more elements to the end of an Array. Using this method, we can append multiple elements by passing multiple arguments.

Question: What is the unshift() method in JavaScript?

The unshift() method is like push(), except it acts on the beginning of the array. This method is used to prepend one or more elements to the beginning of the array.

Question: Explain the pop()method in JavaScript?

The pop() method is similar as the shift() method except that it acts on the last element of the array. Calling pop() on an array will return the last element.

Question: What are the different types of errors in JavaScript?

As of ECMAScript 2016, there are seven other core error constructors in JavaScript (in addition to the generic Error object):

EvalError
An error that occurs during the global function eval().

InternalError
An error that occurs when an internal error in the JavaScript engine is thrown, for example, "too much recursion".

RangeError
An error that occurs when a numeric variable or parameter is outside of its valid range.

ReferenceError
An error that occurs when de-referencing an invalid reference.

SyntaxError
A syntax error that occurs while parsing code in eval().

TypeError
An error that occurs when a variable or parameter is not of a valid type.

URIError
An error that occurs when encodeURI() or decodeURI() are given invalid parameters.

Question: What are JavaScript Cookies?

Cookies are the small text files stored in the computer. Cookies are created when the user visits the websites and are used to store user information that the website might need, such as shopping cart details.

JavaScript can create, read, and delete cookies with the **document.cookie** property.

Question: What is the ‘Strict’ mode in JavaScript and how can it be enabled?

Answer:
ECMAScript 5's strict mode is a way to *opt in* to a restricted variant of JavaScript. Strict mode is intentionally different from normal JavaScript code.

Strict mode makes several changes to normal JavaScript semantics. First, strict mode forces some JavaScript silent errors by changing them to throw errors. Second, strict mode enables JavaScript engines to perform certain optimizations that were previously not possible. Third, strict mode prohibits syntax likely to be defined in future versions of ECMAScript.

Question: Explain closures in JavaScript? When are they used?

A closure is an inner function that has access to the enclosing function's variables. The closure has access to its own scope (variables defined between its curly brackets), access to the outer function's variables, and also it has access to the global variables.

You create a closure by adding a function inside another function. The inner function has access to both the outer functions variables and parameters.

```
function sayHi(firstName, lastName) {
        var greeting = "Hello ";

         // inner function has access to outer
        // function's parameters and variables!

         function name() {
                return greeting + firstName + " "+ lastName;
           }

        return name();
}
```

Question: How are object properties assigned?

Properties are the values associated with a JavaScript object.

Assuming an object already exists, you can simply add a new property like so:
myObject.myPropery = "Property Value";

Question: How can a value be appended to an array?

This is a subtly tricky question because there is the basic answer and a better, more complete answer.

At first glance, you might simply answer that values can be added using array index notation:

```
myArray[myArray.length] = arrayItem;
```

But for a better answer, don't forget all of the array methods that are available!

1. Using *push()* to add to the end of an array
2. Using *unshift()* to add to the beginning of an array
3. Using *splice()* to add elements within an array
4. Using *concat()* to combine and create new, longer arrays

Question: Explain the for-in loop?

The for-in loop is used to loop through the properties of an object. The loop is continued till all the properties of the object are depleted.

```
var item;
for (item in myObject) {
        // do something
}
```

Question: Describe the properties of an anonymous function in JavaScript?

A function that is declared without a name is known as an anonymous function.

Functions stored in variables do not need function names. They are instead called using the variable name.

Question: What logical operators can be used in JavaScript?

The “and”, “or” and “not” operators can be used in JavaScript.

For example, given x=2 and y=6

Operator	Name	Example
&&	and	(x < 3 && y > 1) is true
||	or	(x == 5 || y == 5) is false
!	not	!(x == y) is true

JavaScript and UI Elements

These JavaScript questions focus on elements of the web and user interface.

Question: How can the class of an element be changed?

Using vanilla JavaScript:

```
document.getElementById("MyElement").classList.add('MyClass');

document.getElementById("MyElement").classList.remove('MyClass');
```

Question: How can the style of an element be changed?

Using vanilla JavaScript:

```
document.getElementById("p2").style.color = "blue";
```

Question: Explain how can you submit a form using vanilla JavaScript?

To submit a form using JavaScript use:
document.form[0].submit();

Question: What is the disadvantage of using innerHTML in JavaScript?

Improper handling of the innerHTML property can enable script-injection attacks on Internet Explorer when the HTML string contains a script tag marked as deferred:
<script defer>...<script>

Setting innerHTML will destroy existing HTML elements that have event handlers attached to them, potentially creating a memory leak on some browsers.

You can't set the innerHTML property on all HTML elements on all browsers (for instance, Internet Explorer won't let you set the innerHTML property of a table row element).

Finally, innerHTML content is refreshed every time and thus is slower.

Question: What are all the types of Pop up boxes available in JavaScript?

JavaScript has three kind of popup boxes: Alert box, Confirm box, and Prompt box.

Alert Box
An alert box is often used if you want to make sure information comes through to the user.
When an alert box pops up, the user will have to click "OK" to proceed.

Confirm Box
A confirm box is often used if you want the user to verify or accept something.
When a confirm box pops up, the user will have to click either "OK" or "Cancel" to proceed.

Prompt Box
A prompt box is often used if you want the user to input a value before entering a page.

When a prompt box pops up, the user will have to click either "OK" or "Cancel" to proceed after entering an input value.

Question: What is the use of blur function?

The blur function is used to remove the focus from the specified object.

Question: How do you get the status of a CheckBox?

Using the ID of your input field, you can check the value of:
document.getElementById("my-checkbox").checked

If it is True, the checkbox is checked.

Question: Define event bubbling?

When an event happens on an element, it first runs the handlers on it, then on its parent, then all the way up on other ancestors. In other words, it “bubbles” up.

This is because JavaScript allows DOM elements to be nested inside each other. In such a case, if the handler of the child is clicked, the handler of parent will also work as if it were clicked too.

Question: How are event handlers utilized in JavaScript?

Answer:
Events are the actions that result from activities, such as clicking a link or filling a form, by the user. An **event listener or handler** is a procedure or function in a computer program that waits for an **event** to occur, then performs actions in response to that event.

Compare and Contrast

This section focuses on questions that ask you to explain the similarities and differences between concepts and methodology.

Question: Enumerate the differences between Java and JavaScript?

Although the names are much alike, these two languages are not at all the same and are designed for very different intents. JavaScript is primarily a scripting language for use within HTML pages, while Java is a compiled, programming language that is concurrent, class-based, and object-oriented.

Java was developed by Sun for use as a general-purpose computer programming language.

JavaScript was developed by Netscape, originally as a client side scripting language.

Question: Between JavaScript and an ASP script, which is faster?

JavaScript is faster. JavaScript is a client-side language and thus it does not need the assistance of the web server to execute. On the other hand, ASP is a server-side language and hence is always slower than JavaScript.

Question: What is the difference between ViewState and SessionState?

ViewState is specific to a page in a session.

SessionState is specific to user specific data that can be accessed across all pages in the web application.

Question: Explain the difference between “==” and “===”?

“==” checks only for equality in value.

“===” is a stricter equality test and returns false if either the value or the type of the two variables are different.

Question: What is the difference between an alert box and a confirmation box?

An alert box displays only one button which is the OK button.
A confirmation box displays two buttons, the OK button and the cancel button.

Question: Explain window.onload versus onDocumentReady?

load is called when all assets are done loading, including images. At this point, all of the objects in the document are in the DOM

ready is fired when the DOM is ready for interaction and its elements can be manipulated by your code.This means it will run a little faster than waiting for **load** .

Bonus Questions

These sneaky questions force you to know your JavaScript inside and out. Watch out for tricks!

Question: Is it possible to break JavaScript code into several lines?

Breaking within a string statement can be done by the use of a backslash, '\', at the end of the first line. If f you change to a new line when not within a string statement, then JavaScript ignores break in line.

Question: What would be the result of 3+2+”7″?

Since 3 and 2 are integers, they will be added numerically. And since 7 is a string, its concatenation will be done. So the result would be 57.

Question: How can you convert the string of any base to integer in JavaScript?

The parseInt() function is used to convert numbers between different bases. parseInt() takes the string to be converted as its first parameter, and the second parameter is the base of the given string.

Question: Is JavaScript case sensitive? Give an example?

Yes, JavaScript is case sensitive. For example, a function parseInt() is not same as the function Parseint().

Question: What is the difference between .call() and .apply()?

The function .call() and .apply() are very similar in their usage except that apply() lets you invoke the function with arguments as an array; call() requires the parameters be listed explicitly.In other words, you use apply() when you don't know the number of parameters.