1. Common Expectations
2. Language Syntax

* Variable Naming and declaration
* Data types
* Expression, Statement and Block
  + Variable scope
  + Variable hpoisting
  + keywords
* Operators
  + Arithmetic
  + Comparison
  + Short circuit
  + Nullish Coalescing
  + Optional chaining
  + Ternary
  + Operator Precedence
* Conditional Statements
  + If
  + If else
  + If else if
  + Switch case
* Looping Statement
  + While
  + Do while
  + for
* Functions
  + Declaration and Invocation
  + Function parameter and arguments
  + Default function parameters
  + Using arguments
  + Arrow functions
  + Callback functions
  + Higher order functions
  + Variadic functions
  + IIFE
  + closure
* Strings
  + How String operates
  + Creating a string
  + Getting a character and its code
  + Getting substring
  + String concatenation
  + String template Literal (‘’)
  + String’s utility methods
  + Using regex
* Arrays
  + Creating a Array
  + Array.length
  + Finding index of the element
  + Looping through array
  + Inserting /deleting elements
  + Slice / splice
  + Spread vs rest param
  + Destructing assignment
* Object Literals
  + Creating object through literals
  + JSON vs object literals
  + Getting keys, values
  + Looping through Object entries
  + Spread vs rest param
  + Destructing assignment
* Asynchronous
  + setTimeout , setInterval,requestAnimationFrame
  + Callback Functions
  + Promise API
  + Asynch await
* OOP
  + New operator
  + Constructor function and object creation
  + Getter and setter
  + Class
  + Use of this
  + Prototype object
  + Object inheritance in JS
  + Utility methods of object class

1. Features in JavaScript

* Variable Naming and Declaration
  + Var,let,const
  + Naming Rule
  + Default value
* Data Type , special values
  + Several types
  + Type coercion
  + Typeof , instanceof
  + Undefined,null
  + Nan , Infinity
  + How JS stores numbers

1. Frameworks / Libraries
2. Portfolio / Real Projects
3. Browsers API
   * DOM API
   * Fetch API
   * Graphics
     + Canvas API
     + WebGL
   * Rich Media
     + Web audio API
     + Media Streams API
   * Device
     + Notification API
     + Vibration API
   * Client-side Storage
     + Web storage API
     + IndexedDB API
   * Others
     + Set class
     + Map class
     + Date class
     + Utility Math functions
     + Iterator
     + Generator

[JavaScript Interview Preparation. If you’re applying for a Software… | by Adrian Trujillo Duron | Jan, 2022 | Medium](https://medium.com/@adrian.td96/javascript-interview-preparation-9ba37da8f43)

1. Difference between Java and JavaScript

* It’s a scripting language allows to implement complex features on web pages.
* Displaying timely content updated ,interactive maps,animated graphics,scrolling video jukeboxes.
* It is the third layer in the cke of standard web technologies - HTML and CSS

1. Datatypes supported by JavaScript
2. Difference between let,const and var
3. Difference between == and ===

* == operator will compare both variables irrespective of datatype
* === operator will first check for datatype and then compare the values

Graphical user interface, text, application, email

Description automatically generated

1. Explain promises /closures /callback
2. Difference between undeclared ,undefines and null
3. What is Nan
4. How typeof operator works?
5. What is the use of This operator in Java Script
6. What are arrow functions
7. How to handle exceptions in JavaScript
8. How to create Object in JavaScript
9. Explain OOPS concept in JavaScript
10. Hoisting in JavaScript
11. Ways of embedding JavaScript in HTML file
12. What is temporal Dead Zone ?
13. What are javascript cookies?
14. Object destructing ?
15. What do you mean by ECMAScript?
16. What is escape character
17. Difference between statically and dynamic typed language. Which one in JS?
18. High order function

* It’s a function that operates on functions, taking one more functions as arguments and returning a new function.
* Not () function is high-order function because it takes function argument and return a new function.

Text

Description automatically generated with medium confidence

1. Hoisting?

* Variable and function names can be used before declaring it.
* JavaScript compiler moves all declarations of variables and functions at top so that not be any error.

Text

Description automatically generated

1. Closure ?

* Its combination of function bundled together with references to its surrounding state.
* Closures are created every time a function is created at function creating time.
* You will be able to access the outer function layer , including global scope from inner function.

Graphical user interface, text, application, email

Description automatically generated

1. Currying

* Changing a function havaing multiple arguments into a sequence of functions with single argument.
* Converting the function with more arity into function with less arity (number of parameters in the function)

Benefits :

* + Helps to prevent passing same variable again and again
  + Useful in event handling
  + Define little modules that can be easily reused and configured.

We can achieve currying in 2 methods

* + By using bind() method
  + Using closures
    - Using bind

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

|  |  |
| --- | --- |
| Without currying | With currying |
| <html>  <body>  <script>    function multiply(a, b, c) {  return a \* b \* c;  }  document.write(multiply(2, 5, 8));  </script>  </body>  </html> | <html>  <body>  <script>  function multiply(a) {  return function(b) {  return function(c) {  return a \* b \* c;  }  }  }  document.write(multiply(2)(5)(8));  </script>  </body>  </html> |

1. Arrow Function

* Allows to write shorter function syntax
* They are not suitable for call(),apply(),bind() methods Graphical user interface, application

  Description automatically generated

1. Prototype ?

All JavaScript objects inherit properties and methods from a prototype:

* Date objects inherit from Date.prototype
* Array objects inherit from Array.prototype
* Person objects inherit from Person.prototype

The Object.prototype is on the top of the prototype inheritance chain:

Date objects, Array objects, and Person objects inherit from Object.prototype.

Graphical user interface, text, application

Description automatically generated