

Frontend Interview Bible

This document is broken into 9 Sections.

0. How to use this document.
1. HTML Questions
2. CSS Questions
3. Javascript Conceptual Questions
4. Javascript DOM Based Questions
5. Javascript Task Based Questions
6. Polyfills Questions
7. Machine Coding Questions
8. Frontend Design Questions
9. Performance based Questions
10. How to go beyond this document.

Philosophy behind not adding answers in the document.
Because

1. if we learn from one source, our exploration skills degrade, and our mind becomes lazy. We think that I will read it later and later never comes.
2. Answers given by might not make sense to you. Due to lack of information included which depends on the experience level of someone.
3. If we search for something to learn, our mind retains that thing for a large amount of time.

How to make most of this guide

1. Do not skip any section, Once you complete a section only move to the next section.

2. How to search answer for any question:

Step 1: Copy the question.

Step 2: Open google.com

Step 3: Paste the question in search bar

Step 4: Open first 5 Links in the new tab. (This will prepare you to foresee all possible concepts for any particular topic)

Step 5: Read from all 5 links (this is mandatory)

Step 6: After reading, write the summary anywhere (notes, online blogs wherever you like)

Step 7: Practice code examples at least 3 examples of any concepts.

HTML

1. What does H, T, M , L stand for in HTML and explain each word in detail.
2. Is HTML case sensitive or case in-sensitive?
3. Explain **<head>** tag.
4. How can I **refresh** my page every 60 seconds using HTML?
5. What are **inline** and **block** elements?
6. What is a table tag? And what is the importance of table tag.
7. What are different ways to take **input** from users?
8. How can you make an **upload file** button?
9. How can you **restrict** users to only select a .jpg file when a file upload picker is opened?
10. What does **rowspan** and **colspan** do?
11. Explain **Marquee** tag and its usage.
12. Semantic and Non Semantic Elements.
13. What is the significance of the **<noscript>** tag?
14. Describe the importance of the **<doctype>** declaration in HTML.
15. What is a picture tag and its advantages?
16. What is **srcset** attribute?
17. Explain loading=lazy and loading =eager
18. How can you change the image in the title bar?

CSS

1. What is CSS?
2. What is **inheritance** in CSS?
3. Explain **specificity** rules in CSS?
4. Explain **4 ways** to import CSS in an HTML file.
5. Can I use <style> tag in body instead of Head?
6. Explain **!important**. Can we use it !important in all CSS rules
7. How can you center a div in a webpage, horizontally and vertically. **Give 3 Possible ways.**
8. Explain **Positioning** properties.
9. What is the difference between **sticky** and **fixed**?
10. What is difference between opacity:0, visibility:hidden and display:none
11. How can I only make horizontal scroll, and disable vertical scroll.
12. Explain box model.
13. What does **calc()** function do?
14. Explain **rem** vs **em**.
15. What is the default value of rem.
16. Explain how I can change 1rem = 20px.
17. What are the benefits of using **rem**.
18. Explain **clamp** function

19. How can I show ... when some text is overflowing a container.
20. How can I **restrict** the number of lines using css.
21. How can I animate height width using css
22. Rotate a div by 45 deg
23. Explain flex, and its properties.
24. How can I change the **cursor** using CSS.
25. How can I apply CSS to the first word of a paragraph using CSS.
26. How can I check if user devices support hover or not, and based on that apply CSS.
27. Explain **sibling** selectors.
28. Explain **clearfix** property.
29. Explain how z-index works in CSS.
30. What is the maximum number the z-index supports?
31. Explain background property.
32. What are different measurement units in CSS. Give at least 6 units.
33. Explain what does :has to in CSS.
34. What are vendor prefixes in CSS.
35. How can you hide a scrollbar in CSS.
36. What CSS properties you apply on a button which is disabled.
37. What are pseudo elements?

38. How can you give a border only to the left and right of a div.
39. Difference between outline and border.
40. Explain margin padding collapsing feature.

Javascript

Conceptual Questions:

1. Explain what is Javascript?
2. Is Javascript script single threaded or multithreaded. Explain in detail.
3. How can I add Javascript in my HTML code, give me 2 possible ways.
4. Can I add a script tag in the head element?
5. Does adding script tag in head element cause any problem?
6. What are async and defer attributes?
7. Explain data types in Javascript.
8. What is hoisting? Give examples.
9. What is closure? Give Examples
10. What is stale Closure. Give Examples
11. Explain currying, it's use case and its benefits
12. What is an IIFE?
13. Give 4 ways to create an IIFE.
14. What are operators in JS.
15. How can I check data type of variables in JS.

16. Explain Abstract Equality in Javascript. By Abstract Equality I mean ==.
17. How == is different from ===
18. Explain Execution Context
19. What is Lexical Scope
20. Explain this keyword
21. What is call bind and apply?
22. Explain what is the new Keyword.
23. What are prototypes in JS?
24. Explain difference between __proto__, prototype, [[prototype]]
25. What is prototypical inheritance? How it is different from classical inheritance.
26. What is the difference between var, let and const?
27. What are Symbols in JS?
28. What is the spread and rest concept in JS.
29. Explain destructuring.
30. Explain difference between || and ??
31. What is optional chaining?
32. What is functional programming?
33. What is the difference between reduce and reduceRight.
34. What are callbacks?
35. Explain inversion of control.
36. What are promises?

37. What are the different states of promises?
38. When does the finally block get called under promises?
39. How can I handle errors using try catch in JS.
40. What is tagged template literal.
41. How does this work inside the arrow function?
42. What is the difference between an arrow and normal function? Explain 7 Differences
43. Explain Sets. How they are different from arrays.
44. What are the arguments keywords present in all functions in javascript.
45. What is “use strict” and what it does.
46. How does the for...of loop differ from the for...in loop?
47. How does the async/await syntax simplify asynchronous code in JavaScript?
48. Explain the difference between map and filter.
49. What Map and Weak Map in Javascript?
50. What are weakRef in Javascript.
51. What is difference between setTimeout and requestAnimationFrame.

DOM Manipulation and Events

1. How does event delegation work in the context of JavaScript event handling?
2. Explain the differences between the DOMContentLoaded and load events.
3. How can you stop event propagation in JavaScript?
3. Describe the purpose of the addEventListener method and how it works.
4. What is the event object, and how is it used in JavaScript event handling?
5. Explain the differences between the onclick attribute and addEventListener for handling click events.
6. Explain the concept of bubbling and capturing in event propagation.
7. How can you simulate a click event using JavaScript?
8. Explain the differences between the keyup, keydown, and keypress events.

9. What is the purpose of the preventDefault method in JavaScript events?
10. Difference between querySelector and document.getElementsByClassName

Task based FE Design Question

These questions can be first solved using plain javascript. Then using any JS Library.

In such a question the interviewer tests the skills like how do you create components, what names you use and what folder structure you follow.

1. Find the time taken by a function in Javascript.
2. Implement Infinite Scroll Using Javascript
3. How can you clear all timeouts in Javascript?
4. Print numbers from 1 - 100 with a delay of 2 sec, and also Update the numbers in UI.
5. Show a dialog, when I click a button, make sure to handle the keyboard close, and outside close of dialog.
6. Make an expanded collapse div. (Accordion Implementation). Add relevant Animations.

7. How will you implement a star rating UI?
8. Create an input text, and show suggestions as user types.
Create your own array of suggestions.
9. In The webpage, add background:yellow, to all text which are greater than 5 characters.
10. Implement a Kanban Board using Javascript.
11. You have an api, which gives some data, now you have shown all the data in the UI. Use any only hosting provider, host your own JSON and consume that JSON from your JS Code.
12. Build a calculator with basic arithmetic operations. Use only Plain JS.
13. Create a stopwatch.
14. Create a timer, once time is over show an alert.
15. Demonstrate usage of prompt and confirm box.
16. Using prompt take number from user, and based on that number print a triangle of *. Number given by the user denotes the number of rows in a triangle.
17. Move a div from left of the screen to the right of the screen.
18. Show a tooltip, on hover of a button. Make sure to give the user the ability to change tooltip position. Like left right top bottom.

19. Create a feature using which I can show notifications on my website. Like success, failure , info and warning. Just like a toast message.

Notification should be auto dismissed after 3 seconds.

There can be more than 1 notification in the screen at same time

20. Create tic-tac-toe game

21. How can you lazy load an image?

Polyfills

Polyfills are simply functions that are added on `prototype` of Constructor class, so that they can be available on all instances of the objects created by that particular constructor.

The purpose of polyfill is to support older browsers. Add new functionality in older browsers.

These days, in almost every interview you will encounter a question which demands to write a polyfill.

Below is the list of commonly asked polyfills by interviewers. There are more than 95% chances that if someone asks you to write a polyfill, that polyfill will be from the list below or must be related to the list below.

1. forEach polyfill.
2. Map polyfill.
3. Filter polyfill
4. Reduce polyfill
5. Some and Every function polyfill
6. Push and Pop polyfill
7. Slice and Splice polyfill
8. Call Bind and Apply polyfill
9. Promise.all polyfill
10. Debounce polyfill
11. Throttle polyfill
12. JSON.stringify polyfill

Machine Coding

1. You've given an array which contains some 0, your to write a function which moves all 0s to the right of the array.
2. You've an array in which each adjacent number is either +1 or -1, write a function to find the given number. Focus on efficiency.

Example: [4,5,6,7,6,5,4] // Find index of 7 (from starting)

3. Write a function to shuffle an array.
4. Write a function which gives us a fibonacci sequence of a given number.
5. You have an array, which contains some bad numbers, bad numbers are denoted with -1, write a function to remove all bad numbers from the array.
6. Write a function to flatten a given array.
7. You have a string separated by a . (dot), write a function which converts it into an object. Last character will be added as a value

Example: a.b.c.d.e output => {a : b: { c: { d: e} } }

8. Write a function which merges to given array.
9. Write a function which removes duplicates from array.
10. Write a function which converts a given function into curry version of it.

Example function sum(a,b) {} I should be able to call it like this: sum(2)(2).

11. Given an array containing n distinct numbers taken from 0, 1, 2, ..., n, find the one that is missing.
12. Write a function which deep-clones a given object.
13. Write a function which deep-compares passed value.
14. Write a function which implements ==

15. Write a function which works like a new Keyword.
16. Write a function which makes a chain of promise
17. Write a function which validates if a given set of parentheses are valid or not.
18. Implement a basic string compression algorithm. For example, the string "aaabbbcc" should be compressed to "a3b3c2".

Performance based advanced Questions

Bar Raiser Questions

1. How will you decrease the initial load time?
2. What is CDN? Explain benefits.
3. What are the different cache mechanisms provided by browsers?
4. What is http only cookie.
5. What is CORS?
6. How will you handle 10000 dom elements?
7. What is DNS lookup?
8. What is preload, prefetch and pre connect.
9. Explain SSR, ISR, and SSG
10. What are compression strategies?
11. What is DNS lookup?
12. How will you prevent XSS Attack?
13. How will you prevent CSRF Attack?
14. What is SSE and WebSocket?
15. What are different font loading strategies?
16. What are screen readers?
17. How can you make your website accessible for disabled person?
18. Explain aria attributes.

19. Explain Critical Rendering path.
20. What are 4 Web Vitals?
21. Explain package.json and package.lock JSON
22. What is Tree Shaking?
23. Explain SEO. And how can you accommodate it in SPA?
24. What is Robot.txt file?
25. Explain sitemap.xml file.
26. Explain differences between a production build and a development build.
27. What is SourceMap file.

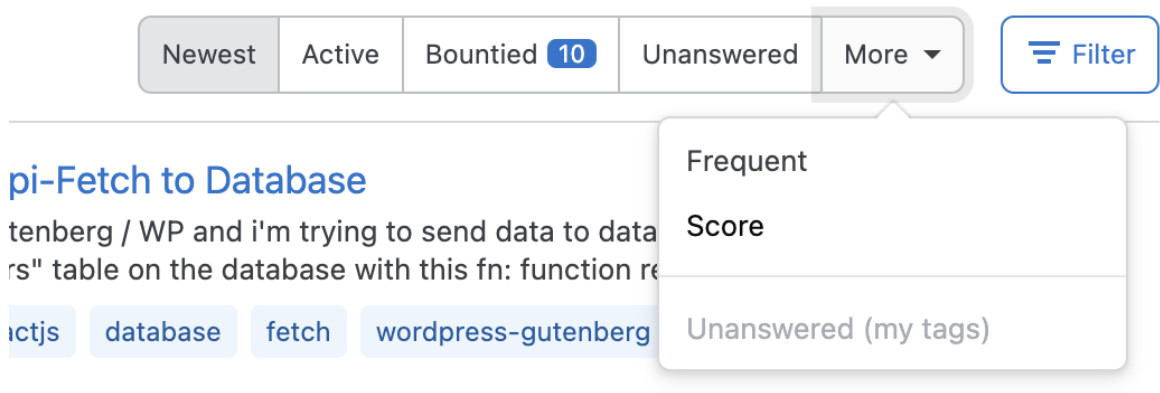
How to go beyond this document.

Tip 1

1. Go to <https://stackoverflow.com/tags>
2. Search your tag : example javascript
3. Select the tag.
4. Sort questions based on score

is NOT Java. Include all tags that are relevant to your question: e.g., [node.js], [laravel], [ember.js], [vue.js], [typescript], [svelte], etc.

1s (14)



custom context menu on textarea if word highlighted using

5. And now you have an infinite list of amazing questions which will take eternity. Now enjoy the learning.

Tip 2

Go to <https://medium.com/tag/javascript>

Now you know the drill, simply sort by newest or popularity and start learning.

JavaScript Concepts:

JavaScript:

<https://github.com/micromata/awesome-javascript-learning> (Ocean of Information)

<https://github.com/topics/javascript-resources>

<https://github.com/Asabeneh/30-Days-Of-JavaScript> - Nice to follow

Basic JavaScript Concepts:

1. Functions

- [Understanding Functions] (<https://www.youtube.com/watch?v=HkWxvB1RJq0>)
- [Deep Dive into JavaScript Functions](<https://www.youtube.com/watch?v=SHINoHxvTso>)

2. Scope in JavaScript

- [Understanding JavaScript Scope](<https://www.youtube.com/watch?v=TkFN6e9ZDMw>)
- [Mastering JavaScript Scope](<https://www.youtube.com/watch?v=uH-tVP8MUs8>)

3. Closure

- [Understanding JavaScript Closure](<https://www.youtube.com/watch?v=qikxEIxsXco>)

4. Event Loop, Call Stack, Task Queue

- [Event Loop in JavaScript](<https://www.youtube.com/watch?v=8aGhZQkoFbQ>)
- [Understanding Call Stack and Event Loop](<https://www.youtube.com/watch?v=8zKuNo4ay8E>)

5. Prototype and Prototype Chain

- [JavaScript Prototype and Inheritance](<https://www.youtube.com/watch?v=wstwjQ1yqWQ>)

6. "THIS" Keyword

- [Demystifying JavaScript "THIS"](<https://www.youtube.com/watch?v=jTPt5XO7KoU>)
- [Understanding JavaScript "THIS"](<https://www.youtube.com/watch?v=YOl79NaAtQ>)
- [In-depth Exploration of "THIS"](<https://www.youtube.com/watch?v=rv7Q11KWmKU>)

7. Class and Inheritance

- [Understanding JavaScript Classes](https://www.youtube.com/watch?v=NyCIWddeO_A)
- [JavaScript Classes and Inheritance](<https://www.youtube.com/watch?v=wstwjQ1yqWQ>)

8. DOM Manipulation

- [Manipulating the DOM with JavaScript](<https://www.youtube.com/watch?v=y17RuWkWdn8>)

9. Bind/Call/Apply

- [Mastering JavaScript Bind, Call, and Apply](<https://www.youtube.com/watch?v=75W8UPQ5I7k>)

10. Promises

- [JavaScript Promises Explained](<https://www.youtube.com/watch?v=ap-6PPAuK1Y>)
- [Deep Dive into Promises](<https://www.youtube.com/watch?v=U74BJcr8NeQ>)

11. Web APIs

- [Exploring JavaScript Web APIs](<https://www.javascripttutorial.net/web-apis/>)

12. Async/Await, Generators

- [Understanding Async/Await](<https://www.youtube.com/watch?v=SrNQS8J67zc>)

13. Functional Programming

- [Introduction to Functional Programming](https://www.youtube.com/watch?v=e-5obm1G_FY)

14. JavaScript Engine

- [Inside the JavaScript Engine](<https://www.youtube.com/watch?v=p-iiEDtpy6I>)
- [JavaScript Engine Fundamentals](<https://www.youtube.com/watch?v=2WJL19wDH68>)

15. How Browser Works

- [Understanding How the Browser Works](<https://www.youtube.com/watch?v=0IsQqJ7pwhw>)

CSS Concepts:

1. Class and Other Selectors

- [Selectors in CSS](<https://www.youtube.com/watch?v=a69OMEJXaJo>)

2. Pseudo Classes and Elements

- [Understanding CSS Pseudo Classes](<https://www.codecademy.com/resources/docs/css/pseudo-classes>)

- [CSS Pseudo Classes Cheatsheet](<https://blog.webdevsimplified.com/2021-12/css-selectors/>)

- [CSS Pseudo Elements](<https://dynamicbalaji.medium.com/css-pseudo-classes-cheatsheet-d336ee537ea0>)

4. Box Model

- [Understanding CSS Box Model](<https://www.youtube.com/watch?v=QyrYEmNYBk>)
- [Mastering CSS Box Model](<https://www.youtube.com/watch?v=rIO5326FgPE>)

5. Flex

- [CSS Flexbox](<https://www.youtube.com/watch?v=fYq5PXgSsbE>)
- [Flexbox Guide](<https://flexbox.malven.co/>)

6. Grid

- [CSS Grid](<https://www.youtube.com/watch?v=0xMQfnTU6oo>)
- [Grid Guide](<https://grid.malven.co/>)

7. Centering Elements

- [CSS Centering Techniques](<https://flaviocopes.com/css-centering/>)

8. Media Queries

- [Responsive Web Design with Media Queries](<https://www.youtube.com/watch?v=69lbzTWg5PM>)

9. CSS Preprocessors - SCSS or LESS

- [Introduction to CSS Preprocessors](<https://www.youtube.com/watch?v=69lbzTWg5PM>)

React Concepts:

1. Component State and Props

- [Understanding React State and Props](<https://www.youtube.com/watch?v=kHJSNfU7H4U>)

2. Styling Components with CSS

3. Functional and Class Components

- [Introduction to React Components](<https://www.youtube.com/watch?v=yc6elaGOoGQ>)

4. React Lifecycle Methods

- [Understanding React Lifecycle Methods](<https://www.youtube.com/watch?v=qoN5FBtfkXs>)

5. React Rendering Process

- [Exploring React Rendering](<https://www.youtube.com/watch?v=i793Qm6kv3U>)

6. Virtual DOM

- [Understanding React Virtual DOM](<https://www.youtube.com/watch?v=rysTbzKOE00>)

7. React Hooks

- `useEffect`: [Mastering useEffect](<https://www.youtube.com/watch?v=UVhIMwHDS7k>)
- `useState`: [Understanding useState](<https://www.youtube.com/watch?v=kkuq0gTGRFQ>)
- `useRef`: [Demystifying useRef](<https://www.youtube.com/watch?v=yviJikU4gwk>)
- `useCallback`: [Understanding
useCallback](<https://www.youtube.com/watch?v=N62nXHTydpA>)
- `useMemo`: [Mastering useMemo](<https://www.youtube.com/watch?v=qGGu46ZoMqQ>)
- `useReducer`: [Deep Dive into
useReducer](<https://www.youtube.com/watch?v=aM1bxz-82Qw>)
- [More on useReducer](<https://www.youtube.com/watch?v=cDdGGthGA6M>)

8. Custom Hooks

- [Creating Custom React Hooks](<https://www.youtube.com/watch?v=nshyjAplovo>)
- [Mastering Custom React Hooks](<https://www.youtube.com/watch?v=NZEUDJvpQMM>)

QUESTIONS

QUESTIONS THAT
YOU WILL BE
ASKED IN INTERVIEW,
AND HOW TO
TACKLE THEM

FRONTEND MASTER - RAHUL

QUESTIONS #1

Q Why don't you introduce your self.

A Avoid traditional intros. Introduce yourself with a story.

Start Like this,

I'm Naren. Let me share a quick story about myself. Since childhood, I've enjoyed solving puzzles, like figuring out the knots in my mother's biscuit bags. Little did I know this curiosity would lead me to my passion for debugging code and fixing bugs in programming.

After school, I joined XYZ College and started my career in 20XX as a XYZ Developer. Now, let me give you a glimpse into my typical workday and highlight some of my technical skills

QUESTIONS #2

Q Is there any application/website/work that you are proud of?

A By asking such questions, sometimes interviewers want to check how you explain your work, and if you know what you have written in your resume or not.

Use this question as an opportunity and explain the best work you have ever done.

Prepare for such questions before interview, and be ready for cross questioning.

QUESTIONS #3

Q What is the biggest problem you have faced and how did you solved it?

A Answer of this questions makes it clear the scale at which you have ever worked.

If you quote a simple problem it might backfire.

So use this question as an oppprtunity to showcase your skills, simply explain a extreme tough situation even tough you haven faced, but you know how to solve it.

Be prepared for cross questioning.

QUESTIONS #4

Q You came to know that your team mate has done a blunder what will you do?

A This question will check your honesty towards work.

You can simply say that you will try to help your friend to resolve that blunder before anyone knows.

And if anytime you know that you can not manage, you will escalate this and let your manager know.

QUESTIONS #5

Q You have give a problem which is above your role or skills. What will you do?

A This question will check how will you behave when put on tough spot, simply tell them I've assigned such work I will love to learn new skills considering I've give enough time to learn.

Otherwise I will take some help from my seniors, and if no help is available, and the task is critical I will spend extra time to learn and fix it.

QUESTIONS #6

Q You have applied for XYZ, Do you know what XYZ do?

A Make sure to know the basics of the company in which you are interviewing.

When HR calls its ok to say that you don't know.

But once your interview is scheduled, and the interviewer asks this, do you know what we do?

If you say NO, or even say that you only know this and this, that won't go in your favour.

Instead you can say whatever you have prepared, and ask the interviewer that you will love to hear more if you have missed anything,

QUESTIONS #7

Q Suppose you are working on a task, and you got to know another issue what will you do?

A This question checks if you are ready to take ownership or not.

Simply say if that issue is small you will fix it along with current task, and if that issue is taking time like more than 1 hour, you will report it and continue on your current task.

ALL THE BEST FOR YOUR NEXT INTERVIEW

FRONTEND MASTER - RAHUL

Frontend Awareness Must Known Things

1. Explain What is DNS? And How it Works?
2. Explain the Difference Between Authentication and Authorization.
3. What is the Purpose of a Session Token in Web Security?
4. Explain the Role of JSON Web Tokens (JWT) in Authentication.
5. What is TCP?
6. Describe the Three-Way Handshake in TCP.
7. What is SSL?
8. How client server works?
9. How will you deploy your application to server?
10. When you press enter in addressbar. Explain whole process from pressing enter to showing content on browser.
11. What is SEO?
12. What is difference between SPA and traditional Web apps.
13. What is an API?
14. OOPS Concepts.
15. Race Conditions, Deadlocks.
16. Parallel vs Concurrency in programming
17. What do you understand by CMS.
18. What is HTTP / HTTP 2.
19. Explain REST.
20. How does Router works.
21. Explain LAN MAN and WAN
22. Web Security - How to you prevent security attacks
23. Explain the Role of a Firewall in Network Security.
24. What is CI/CD
25. Git and GitHub
26. Exposure to any design tool like Figma (Not the designing but should be aware about how to use)
27. What is pixel?

Frontend Interview Bible

This document is broken into 9 Sections.

0. How to use this document.
1. HTML Questions
2. CSS Questions
3. Javascript Conceptual Questions
4. Javascript DOM Based Questions
5. Javascript Task Based Questions
6. Polyfills Questions
7. Machine Coding Questions
8. Frontend Design Questions
9. Performance based Questions
10. How to go beyond this document.

Philosophy behind not adding answers in the document.
Because

1. if we learn from one source, our exploration skills degrade, and our mind becomes lazy. We think that I will read it later and later never comes.
2. Answers given by might not make sense to you. Due to lack of information included which depends on the experience level of someone.
3. If we search for something to learn, our mind retains that thing for a large amount of time.

How to make most of this guide

1. Do not skip any section, Once you complete a section only move to the next section.

2. How to search answer for any question:

Step 1: Copy the question.

Step 2: Open google.com

Step 3: Paste the question in search bar

Step 4: Open first 5 Links in the new tab. (This will prepare you to foresee all possible concepts for any particular topic)

Step 5: Read from all 5 links (this is mandatory)

Step 6: After reading, write the summary anywhere (notes, online blogs wherever you like)

Step 7: Practice code examples at least 3 examples of any concepts.

HTML

1. What does H, T, M , L stand for in HTML and explain each word in detail.
2. Is HTML case sensitive or case in-sensitive?
3. Explain **<head>** tag.
4. How can I **refresh** my page every 60 seconds using HTML?
5. What are **inline** and **block** elements?
6. What is a table tag? And what is the importance of table tag.
7. What are different ways to take **input** from users?
8. How can you make an **upload file** button?
9. How can you **restrict** users to only select a .jpg file when a file upload picker is opened?
10. What does **rowspan** and **colspan** do?
11. Explain **Marquee** tag and its usage.
12. Semantic and Non Semantic Elements.
13. What is the significance of the **<noscript>** tag?
14. Describe the importance of the **<doctype>** declaration in HTML.
15. What is a picture tag and its advantages?
16. What is **srcset** attribute?
17. Explain loading=lazy and loading =eager
18. How can you change the image in the title bar?

CSS

1. What is CSS?
2. What is **inheritance** in CSS?
3. Explain **specificity** rules in CSS?
4. Explain **4 ways** to import CSS in an HTML file.
5. Can I use <style> tag in body instead of Head?
6. Explain **!important**. Can we use it !important in all CSS rules
7. How can you center a div in a webpage, horizontally and vertically. **Give 3 Possible ways.**
8. Explain **Positioning** properties.
9. What is the difference between **sticky** and **fixed**?
10. What is difference between opacity:0, visibility:hidden and display:none
11. How can I only make horizontal scroll, and disable vertical scroll.
12. Explain box model.
13. What does **calc()** function do?
14. Explain **rem** vs **em**.
15. What is the default value of rem.
16. Explain how I can change 1rem = 20px.
17. What are the benefits of using **rem**.
18. Explain **clamp** function

19. How can I show ... when some text is overflowing a container.
20. How can I **restrict** the number of lines using css.
21. How can I animate height width using css
22. Rotate a div by 45 deg
23. Explain flex, and its properties.
24. How can I change the **cursor** using CSS.
25. How can I apply CSS to the first word of a paragraph using CSS.
26. How can I check if user devices support hover or not, and based on that apply CSS.
27. Explain **sibling** selectors.
28. Explain **clearfix** property.
29. Explain how z-index works in CSS.
30. What is the maximum number the z-index supports?
31. Explain background property.
32. What are different measurement units in CSS. Give at least 6 units.
33. Explain what does :has to in CSS.
34. What are vendor prefixes in CSS.
35. How can you hide a scrollbar in CSS.
36. What CSS properties you apply on a button which is disabled.
37. What are pseudo elements?

38. How can you give a border only to the left and right of a div.
39. Difference between outline and border.
40. Explain margin padding collapsing feature.

Javascript

Conceptual Questions:

1. Explain what is Javascript?
2. Is Javascript script single threaded or multithreaded. Explain in detail.
3. How can I add Javascript in my HTML code, give me 2 possible ways.
4. Can I add a script tag in the head element?
5. Does adding script tag in head element cause any problem?
6. What are async and defer attributes?
7. Explain data types in Javascript.
8. What is hoisting? Give examples.
9. What is closure? Give Examples
10. What is stale Closure. Give Examples
11. Explain currying, it's use case and its benefits
12. What is an IIFE?
13. Give 4 ways to create an IIFE.
14. What are operators in JS.
15. How can I check data type of variables in JS.

16. Explain Abstract Equality in Javascript. By Abstract Equality I mean ==.
17. How == is different from ===
18. Explain Execution Context
19. What is Lexical Scope
20. Explain this keyword
21. What is call bind and apply?
22. Explain what is the new Keyword.
23. What are prototypes in JS?
24. Explain difference between __proto__, prototype, [[prototype]]
25. What is prototypical inheritance? How it is different from classical inheritance.
26. What is the difference between var, let and const?
27. What are Symbols in JS?
28. What is the spread and rest concept in JS.
29. Explain destructuring.
30. Explain difference between || and ??
31. What is optional chaining?
32. What is functional programming?
33. What is the difference between reduce and reduceRight.
34. What are callbacks?
35. Explain inversion of control.
36. What are promises?

37. What are the different states of promises?
38. When does the finally block get called under promises?
39. How can I handle errors using try catch in JS.
40. What is tagged template literal.
41. How does this work inside the arrow function?
42. What is the difference between an arrow and normal function? Explain 7 Differences
43. Explain Sets. How they are different from arrays.
44. What are the arguments keywords present in all functions in javascript.
45. What is “use strict” and what it does.
46. How does the for...of loop differ from the for...in loop?
47. How does the async/await syntax simplify asynchronous code in JavaScript?
48. Explain the difference between map and filter.
49. What Map and Weak Map in Javascript?
50. What are weakRef in Javascript.
51. What is difference between setTimeout and requestAnimationFrame.

DOM Manipulation and Events

1. How does event delegation work in the context of JavaScript event handling?
2. Explain the differences between the DOMContentLoaded and load events.
3. How can you stop event propagation in JavaScript?
3. Describe the purpose of the addEventListener method and how it works.
4. What is the event object, and how is it used in JavaScript event handling?
5. Explain the differences between the onclick attribute and addEventListener for handling click events.
6. Explain the concept of bubbling and capturing in event propagation.
7. How can you simulate a click event using JavaScript?
8. Explain the differences between the keyup, keydown, and keypress events.

9. What is the purpose of the preventDefault method in JavaScript events?
10. Difference between querySelector and document.getElementsByClassName

Task based FE Design Question

These questions can be first solved using plain javascript. Then using any JS Library.

In such a question the interviewer tests the skills like how do you create components, what names you use and what folder structure you follow.

1. Find the time taken by a function in Javascript.
2. Implement Infinite Scroll Using Javascript
3. How can you clear all timeouts in Javascript?
4. Print numbers from 1 - 100 with a delay of 2 sec, and also Update the numbers in UI.
5. Show a dialog, when I click a button, make sure to handle the keyboard close, and outside close of dialog.
6. Make an expanded collapse div. (Accordion Implementation). Add relevant Animations.

7. How will you implement a star rating UI?
8. Create an input text, and show suggestions as user types.
Create your own array of suggestions.
9. In The webpage, add background:yellow, to all text which are greater than 5 characters.
10. Implement a Kanban Board using Javascript.
11. You have an api, which gives some data, now you have shown all the data in the UI. Use any only hosting provider, host your own JSON and consume that JSON from your JS Code.
12. Build a calculator with basic arithmetic operations. Use only Plain JS.
13. Create a stopwatch.
14. Create a timer, once time is over show an alert.
15. Demonstrate usage of prompt and confirm box.
16. Using prompt take number from user, and based on that number print a triangle of *. Number given by the user denotes the number of rows in a triangle.
17. Move a div from left of the screen to the right of the screen.
18. Show a tooltip, on hover of a button. Make sure to give the user the ability to change tooltip position. Like left right top bottom.

19. Create a feature using which I can show notifications on my website. Like success, failure , info and warning. Just like a toast message.

Notification should be auto dismissed after 3 seconds.

There can be more than 1 notification in the screen at same time

20. Create tic-tac-toe game

21. How can you lazy load an image?

Polyfills

Polyfills are simply functions that are added on `prototype` of Constructor class, so that they can be available on all instances of the objects created by that particular constructor.

The purpose of polyfill is to support older browsers. Add new functionality in older browsers.

These days, in almost every interview you will encounter a question which demands to write a polyfill.

Below is the list of commonly asked polyfills by interviewers. There are more than 95% chances that if someone asks you to write a polyfill, that polyfill will be from the list below or must be related to the list below.

1. forEach polyfill.
2. Map polyfill.
3. Filter polyfill
4. Reduce polyfill
5. Some and Every function polyfill
6. Push and Pop polyfill
7. Slice and Splice polyfill
8. Call Bind and Apply polyfill
9. Promise.all polyfill
10. Debounce polyfill
11. Throttle polyfill
12. JSON.stringify polyfill

Machine Coding

1. You've given an array which contains some 0, your to write a function which moves all 0s to the right of the array.
2. You've an array in which each adjacent number is either +1 or -1, write a function to find the given number. Focus on efficiency.

Example: [4,5,6,7,6,5,4] // Find index of 7 (from starting)

3. Write a function to shuffle an array.
4. Write a function which gives us a fibonacci sequence of a given number.
5. You have an array, which contains some bad numbers, bad numbers are denoted with -1, write a function to remove all bad numbers from the array.
6. Write a function to flatten a given array.
7. You have a string separated by a . (dot), write a function which converts it into an object. Last character will be added as a value

Example: a.b.c.d.e output => {a : b: { c: { d: e} } }

8. Write a function which merges to given array.
9. Write a function which removes duplicates from array.
10. Write a function which converts a given function into curry version of it.

Example function sum(a,b) {} I should be able to call it like this: sum(2)(2).

11. Given an array containing n distinct numbers taken from 0, 1, 2, ..., n, find the one that is missing.
12. Write a function which deep-clones a given object.
13. Write a function which deep-compares passed value.
14. Write a function which implements ==

15. Write a function which works like a new Keyword.
16. Write a function which makes a chain of promise
17. Write a function which validates if a given set of parentheses are valid or not.
18. Implement a basic string compression algorithm. For example, the string "aaabbbcc" should be compressed to "a3b3c2".

Performance based advanced Questions

Bar Raiser Questions

1. How will you decrease the initial load time?
2. What is CDN? Explain benefits.
3. What are the different cache mechanisms provided by browsers?
4. What is http only cookie.
5. What is CORS?
6. How will you handle 10000 dom elements?
7. What is DNS lookup?
8. What is preload, prefetch and pre connect.
9. Explain SSR, ISR, and SSG
10. What are compression strategies?
11. What is DNS lookup?
12. How will you prevent XSS Attack?
13. How will you prevent CSRF Attack?
14. What is SSE and WebSocket?
15. What are different font loading strategies?
16. What are screen readers?
17. How can you make your website accessible for disabled person?
18. Explain aria attributes.

19. Explain Critical Rendering path.
20. What are 4 Web Vitals?
21. Explain package.json and package.lock JSON
22. What is Tree Shaking?
23. Explain SEO. And how can you accommodate it in SPA?
24. What is Robot.txt file?
25. Explain sitemap.xml file.
26. Explain differences between a production build and a development build.
27. What is SourceMap file.

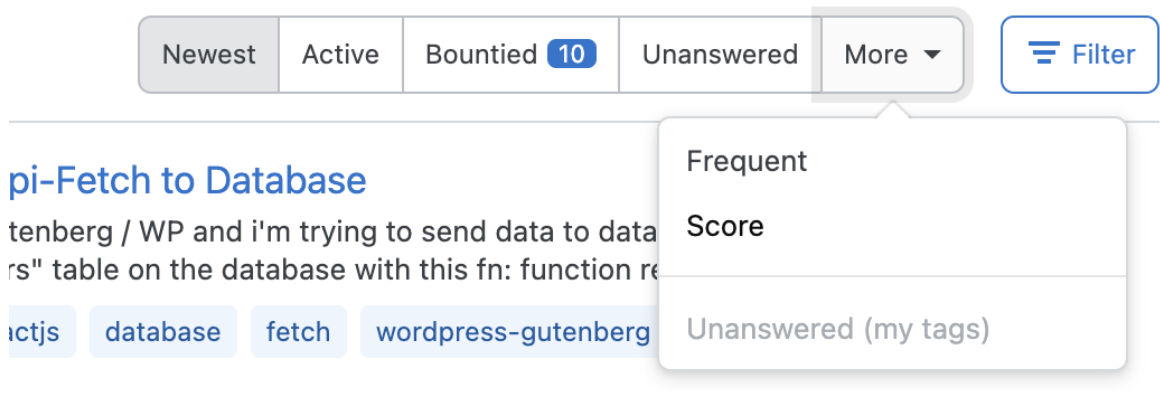
How to go beyond this document.

Tip 1

1. Go to <https://stackoverflow.com/tags>
2. Search your tag : example javascript
3. Select the tag.
4. Sort questions based on score

is NOT Java. Include all tags that are relevant to your question: e.g., [node.js], [laravel], [ember.js], [vue.js], [typescript], [svelte], etc.

1s (14)



custom context menu on textarea if word highlighted using

5. And now you have an infinite list of amazing questions which will take eternity. Now enjoy the learning.

Tip 2

Go to <https://medium.com/tag/javascript>

Now you know the drill, simply sort by newest or popularity and start learning.

Frontend Interview Bible

This document is broken into 9 Sections.

0. How to use this document.
1. HTML Questions
2. CSS Questions
3. Javascript Conceptual Questions
4. Javascript DOM Based Questions
5. Javascript Task Based Questions
6. Polyfills Questions
7. Machine Coding Questions
8. Frontend Design Questions
9. Performance based Questions
10. How to go beyond this document.

Philosophy behind not adding answers in the document.
Because

1. if we learn from one source, our exploration skills degrade, and our mind becomes lazy. We think that I will read it later and later never comes.
2. Answers given by might not make sense to you. Due to lack of information included which depends on the experience level of someone.
3. If we search for something to learn, our mind retains that thing for a large amount of time.

How to make most of this guide

1. Do not skip any section, Once you complete a section only move to the next section.

2. How to search answer for any question:

Step 1: Copy the question.

Step 2: Open google.com

Step 3: Paste the question in search bar

Step 4: Open first 5 Links in the new tab. (This will prepare you to foresee all possible concepts for any particular topic)

Step 5: Read from all 5 links (this is mandatory)

Step 6: After reading, write the summary anywhere (notes, online blogs wherever you like)

Step 7: Practice code examples at least 3 examples of any concepts.

HTML

1. What does H, T, M , L stand for in HTML and explain each word in detail.
2. Is HTML case sensitive or case in-sensitive?
3. Explain **<head>** tag.
4. How can I **refresh** my page every 60 seconds using HTML?
5. What are **inline** and **block** elements?
6. What is a table tag? And what is the importance of table tag.
7. What are different ways to take **input** from users?
8. How can you make an **upload file** button?
9. How can you **restrict** users to only select a .jpg file when a file upload picker is opened?
10. What does **rowspan** and **colspan** do?
11. Explain **Marquee** tag and its usage.
12. Semantic and Non Semantic Elements.
13. What is the significance of the **<noscript>** tag?
14. Describe the importance of the **<doctype>** declaration in HTML.
15. What is a picture tag and its advantages?
16. What is **srcset** attribute?
17. Explain loading=lazy and loading =eager
18. How can you change the image in the title bar?

CSS

1. What is CSS?
2. What is **inheritance** in CSS?
3. Explain **specificity** rules in CSS?
4. Explain **4 ways** to import CSS in an HTML file.
5. Can I use <style> tag in body instead of Head?
6. Explain **!important**. Can we use it !important in all CSS rules
7. How can you center a div in a webpage, horizontally and vertically. **Give 3 Possible ways.**
8. Explain **Positioning** properties.
9. What is the difference between **sticky** and **fixed**?
10. What is difference between opacity:0, visibility:hidden and display:none
11. How can I only make horizontal scroll, and disable vertical scroll.
12. Explain box model.
13. What does **calc()** function do?
14. Explain **rem** vs **em**.
15. What is the default value of rem.
16. Explain how I can change 1rem = 20px.
17. What are the benefits of using **rem**.
18. Explain **clamp** function

19. How can I show ... when some text is overflowing a container.
20. How can I **restrict** the number of lines using css.
21. How can I animate height width using css
22. Rotate a div by 45 deg
23. Explain flex, and its properties.
24. How can I change the **cursor** using CSS.
25. How can I apply CSS to the first word of a paragraph using CSS.
26. How can I check if user devices support hover or not, and based on that apply CSS.
27. Explain **sibling** selectors.
28. Explain **clearfix** property.
29. Explain how z-index works in CSS.
30. What is the maximum number the z-index supports?
31. Explain background property.
32. What are different measurement units in CSS. Give at least 6 units.
33. Explain what does :has to in CSS.
34. What are vendor prefixes in CSS.
35. How can you hide a scrollbar in CSS.
36. What CSS properties you apply on a button which is disabled.
37. What are pseudo elements?

38. How can you give a border only to the left and right of a div.
39. Difference between outline and border.
40. Explain margin padding collapsing feature.

Javascript

Conceptual Questions:

1. Explain what is Javascript?
2. Is Javascript script single threaded or multithreaded. Explain in detail.
3. How can I add Javascript in my HTML code, give me 2 possible ways.
4. Can I add a script tag in the head element?
5. Does adding script tag in head element cause any problem?
6. What are async and defer attributes?
7. Explain data types in Javascript.
8. What is hoisting? Give examples.
9. What is closure? Give Examples
10. What is stale Closure. Give Examples
11. Explain currying, it's use case and its benefits
12. What is an IIFE?
13. Give 4 ways to create an IIFE.
14. What are operators in JS.
15. How can I check data type of variables in JS.

16. Explain Abstract Equality in Javascript. By Abstract Equality I mean ==.
17. How == is different from ===
18. Explain Execution Context
19. What is Lexical Scope
20. Explain this keyword
21. What is call bind and apply?
22. Explain what is the new Keyword.
23. What are prototypes in JS?
24. Explain difference between __proto__, prototype, [[prototype]]
25. What is prototypical inheritance? How it is different from classical inheritance.
26. What is the difference between var, let and const?
27. What are Symbols in JS?
28. What is the spread and rest concept in JS.
29. Explain destructuring.
30. Explain difference between || and ??
31. What is optional chaining?
32. What is functional programming?
33. What is the difference between reduce and reduceRight.
34. What are callbacks?
35. Explain inversion of control.
36. What are promises?

37. What are the different states of promises?
38. When does the finally block get called under promises?
39. How can I handle errors using try catch in JS.
40. What is tagged template literal.
41. How does this work inside the arrow function?
42. What is the difference between an arrow and normal function? Explain 7 Differences
43. Explain Sets. How they are different from arrays.
44. What are the arguments keywords present in all functions in javascript.
45. What is “use strict” and what it does.
46. How does the for...of loop differ from the for...in loop?
47. How does the async/await syntax simplify asynchronous code in JavaScript?
48. Explain the difference between map and filter.
49. What Map and Weak Map in Javascript?
50. What are weakRef in Javascript.
51. What is difference between setTimeout and requestAnimationFrame.

DOM Manipulation and Events

1. How does event delegation work in the context of JavaScript event handling?
2. Explain the differences between the DOMContentLoaded and load events.
3. How can you stop event propagation in JavaScript?
3. Describe the purpose of the addEventListener method and how it works.
4. What is the event object, and how is it used in JavaScript event handling?
5. Explain the differences between the onclick attribute and addEventListener for handling click events.
6. Explain the concept of bubbling and capturing in event propagation.
7. How can you simulate a click event using JavaScript?
8. Explain the differences between the keyup, keydown, and keypress events.

9. What is the purpose of the preventDefault method in JavaScript events?
10. Difference between querySelector and document.getElementsByClassName

Task based FE Design Question

These questions can be first solved using plain javascript. Then using any JS Library.

In such a question the interviewer tests the skills like how do you create components, what names you use and what folder structure you follow.

1. Find the time taken by a function in Javascript.
2. Implement Infinite Scroll Using Javascript
3. How can you clear all timeouts in Javascript?
4. Print numbers from 1 - 100 with a delay of 2 sec, and also Update the numbers in UI.
5. Show a dialog, when I click a button, make sure to handle the keyboard close, and outside close of dialog.
6. Make an expanded collapse div. (Accordion Implementation). Add relevant Animations.

7. How will you implement a star rating UI?
8. Create an input text, and show suggestions as user types.
Create your own array of suggestions.
9. In The webpage, add background:yellow, to all text which are greater than 5 characters.
10. Implement a Kanban Board using Javascript.
11. You have an api, which gives some data, now you have shown all the data in the UI. Use any only hosting provider, host your own JSON and consume that JSON from your JS Code.
12. Build a calculator with basic arithmetic operations. Use only Plain JS.
13. Create a stopwatch.
14. Create a timer, once time is over show an alert.
15. Demonstrate usage of prompt and confirm box.
16. Using prompt take number from user, and based on that number print a triangle of *. Number given by the user denotes the number of rows in a triangle.
17. Move a div from left of the screen to the right of the screen.
18. Show a tooltip, on hover of a button. Make sure to give the user the ability to change tooltip position. Like left right top bottom.

19. Create a feature using which I can show notifications on my website. Like success, failure , info and warning. Just like a toast message.

Notification should be auto dismissed after 3 seconds.

There can be more than 1 notification in the screen at same time

20. Create tic-tac-toe game

21. How can you lazy load an image?

Polyfills

Polyfills are simply functions that are added on `prototype` of Constructor class, so that they can be available on all instances of the objects created by that particular constructor.

The purpose of polyfill is to support older browsers. Add new functionality in older browsers.

These days, in almost every interview you will encounter a question which demands to write a polyfill.

Below is the list of commonly asked polyfills by interviewers. There are more than 95% chances that if someone asks you to write a polyfill, that polyfill will be from the list below or must be related to the list below.

1. forEach polyfill.
2. Map polyfill.
3. Filter polyfill
4. Reduce polyfill
5. Some and Every function polyfill
6. Push and Pop polyfill
7. Slice and Splice polyfill
8. Call Bind and Apply polyfill
9. Promise.all polyfill
10. Debounce polyfill
11. Throttle polyfill
12. JSON.stringify polyfill

Machine Coding

1. You've given an array which contains some 0, your to write a function which moves all 0s to the right of the array.
2. You've an array in which each adjacent number is either +1 or -1, write a function to find the given number. Focus on efficiency.

Example: [4,5,6,7,6,5,4] // Find index of 7 (from starting)

3. Write a function to shuffle an array.
4. Write a function which gives us a fibonacci sequence of a given number.
5. You have an array, which contains some bad numbers, bad numbers are denoted with -1, write a function to remove all bad numbers from the array.
6. Write a function to flatten a given array.
7. You have a string separated by a . (dot), write a function which converts it into an object. Last character will be added as a value

Example: a.b.c.d.e output => {a : b: { c: { d: e} } }

8. Write a function which merges to given array.
9. Write a function which removes duplicates from array.
10. Write a function which converts a given function into curry version of it.

Example function sum(a,b) {} I should be able to call it like this: sum(2)(2).

11. Given an array containing n distinct numbers taken from 0, 1, 2, ..., n, find the one that is missing.
12. Write a function which deep-clones a given object.
13. Write a function which deep-compares passed value.
14. Write a function which implements ==

15. Write a function which works like a new Keyword.
16. Write a function which makes a chain of promise
17. Write a function which validates if a given set of parentheses are valid or not.
18. Implement a basic string compression algorithm. For example, the string "aaabbbcc" should be compressed to "a3b3c2".

Performance based advanced Questions

Bar Raiser Questions

1. How will you decrease the initial load time?
2. What is CDN? Explain benefits.
3. What are the different cache mechanisms provided by browsers?
4. What is http only cookie.
5. What is CORS?
6. How will you handle 10000 dom elements?
7. What is DNS lookup?
8. What is preload, prefetch and pre connect.
9. Explain SSR, ISR, and SSG
10. What are compression strategies?
11. What is DNS lookup?
12. How will you prevent XSS Attack?
13. How will you prevent CSRF Attack?
14. What is SSE and WebSocket?
15. What are different font loading strategies?
16. What are screen readers?
17. How can you make your website accessible for disabled person?
18. Explain aria attributes.

19. Explain Critical Rendering path.
20. What are 4 Web Vitals?
21. Explain package.json and package.lock JSON
22. What is Tree Shaking?
23. Explain SEO. And how can you accommodate it in SPA?
24. What is Robot.txt file?
25. Explain sitemap.xml file.
26. Explain differences between a production build and a development build.
27. What is SourceMap file.

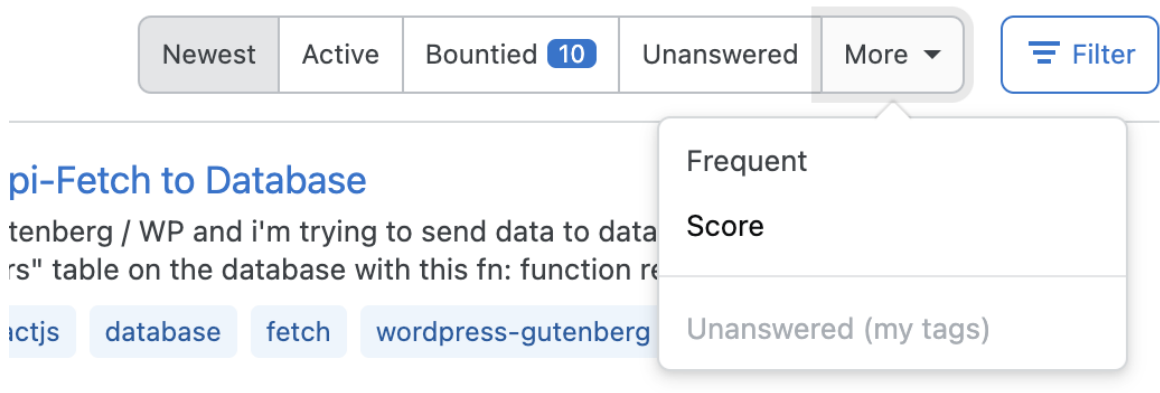
How to go beyond this document.

Tip 1

1. Go to <https://stackoverflow.com/tags>
2. Search your tag : example javascript
3. Select the tag.
4. Sort questions based on score

is NOT Java. Include all tags that are relevant to your question: e.g., [node.js], [laravel], [ember.js], [vue.js], [typescript], [svelte], etc.

1s (14)



custom context menu on textarea if word highlighted using

5. And now you have an infinite list of amazing questions which will take eternity. Now enjoy the learning.

Tip 2

Go to <https://medium.com/tag/javascript>

Now you know the drill, simply sort by newest or popularity and start learning.

JavaScript Concepts:

JavaScript:

<https://github.com/micromata/awesome-javascript-learning> (Ocean of Information)

<https://github.com/topics/javascript-resources>

<https://github.com/Asabeneh/30-Days-Of-JavaScript> - Nice to follow

Basic JavaScript Concepts:

1. Functions

- [Understanding Functions] (<https://www.youtube.com/watch?v=HkWxvB1RJq0>)
- [Deep Dive into JavaScript Functions](<https://www.youtube.com/watch?v=SHINoHxvTso>)

2. Scope in JavaScript

- [Understanding JavaScript Scope](<https://www.youtube.com/watch?v=TkFN6e9ZDMw>)
- [Mastering JavaScript Scope](<https://www.youtube.com/watch?v=uH-tVP8MUs8>)

3. Closure

- [Understanding JavaScript Closure](<https://www.youtube.com/watch?v=qikxEIxsXco>)

4. Event Loop, Call Stack, Task Queue

- [Event Loop in JavaScript](<https://www.youtube.com/watch?v=8aGhZQkoFbQ>)
- [Understanding Call Stack and Event Loop](<https://www.youtube.com/watch?v=8zKuNo4ay8E>)

5. Prototype and Prototype Chain

- [JavaScript Prototype and Inheritance](<https://www.youtube.com/watch?v=wstwjQ1yqWQ>)

6. "THIS" Keyword

- [Demystifying JavaScript "THIS"](<https://www.youtube.com/watch?v=jTPt5XO7KoU>)
- [Understanding JavaScript "THIS"](<https://www.youtube.com/watch?v=YOl79NaAtQ>)
- [In-depth Exploration of "THIS"](<https://www.youtube.com/watch?v=rv7Q11KWmKU>)

7. Class and Inheritance

- [Understanding JavaScript Classes](https://www.youtube.com/watch?v=NyCIWddeO_A)
- [JavaScript Classes and Inheritance](<https://www.youtube.com/watch?v=wstwjQ1yqWQ>)

8. DOM Manipulation

- [Manipulating the DOM with JavaScript](<https://www.youtube.com/watch?v=y17RuWkWdn8>)

9. Bind/Call/Apply

- [Mastering JavaScript Bind, Call, and Apply](<https://www.youtube.com/watch?v=75W8UPQ5I7k>)

10. Promises

- [JavaScript Promises Explained](<https://www.youtube.com/watch?v=ap-6PPAuK1Y>)
- [Deep Dive into Promises](<https://www.youtube.com/watch?v=U74BJcr8NeQ>)

11. Web APIs

- [Exploring JavaScript Web APIs](<https://www.javascripttutorial.net/web-apis/>)

12. Async/Await, Generators

- [Understanding Async/Await](<https://www.youtube.com/watch?v=SrNQS8J67zc>)

13. Functional Programming

- [Introduction to Functional Programming](https://www.youtube.com/watch?v=e-5obm1G_FY)

14. JavaScript Engine

- [Inside the JavaScript Engine](<https://www.youtube.com/watch?v=p-iiEDtpy6I>)
- [JavaScript Engine Fundamentals](<https://www.youtube.com/watch?v=2WJL19wDH68>)

15. How Browser Works

- [Understanding How the Browser Works](<https://www.youtube.com/watch?v=0IsQqJ7pwhw>)

CSS Concepts:

1. Class and Other Selectors

- [Selectors in CSS](<https://www.youtube.com/watch?v=a69OMEJXaJo>)

2. Pseudo Classes and Elements

- [Understanding CSS Pseudo Classes](<https://www.codecademy.com/resources/docs/css/pseudo-classes>)

- [CSS Pseudo Classes Cheatsheet](<https://blog.webdevsimplified.com/2021-12/css-selectors/>)

- [CSS Pseudo Elements](<https://dynamicbalaji.medium.com/css-pseudo-classes-cheatsheet-d336ee537ea0>)

4. Box Model

- [Understanding CSS Box Model](<https://www.youtube.com/watch?v=QyrYEmNYBk>)
- [Mastering CSS Box Model](<https://www.youtube.com/watch?v=rIO5326FgPE>)

5. Flex

- [CSS Flexbox](<https://www.youtube.com/watch?v=fYq5PXgSsbE>)
- [Flexbox Guide](<https://flexbox.malven.co/>)

6. Grid

- [CSS Grid](<https://www.youtube.com/watch?v=0xMQfnTU6oo>)
- [Grid Guide](<https://grid.malven.co/>)

7. Centering Elements

- [CSS Centering Techniques](<https://flaviocopes.com/css-centering/>)

8. Media Queries

- [Responsive Web Design with Media Queries](<https://www.youtube.com/watch?v=69lbzTWg5PM>)

9. CSS Preprocessors - SCSS or LESS

- [Introduction to CSS Preprocessors](<https://www.youtube.com/watch?v=69lbzTWg5PM>)

React Concepts:

1. Component State and Props

- [Understanding React State and Props](<https://www.youtube.com/watch?v=kHJSNfU7H4U>)

2. Styling Components with CSS

3. Functional and Class Components

- [Introduction to React Components](<https://www.youtube.com/watch?v=yc6elaGOoGQ>)

4. React Lifecycle Methods

- [Understanding React Lifecycle Methods](<https://www.youtube.com/watch?v=qoN5FBtfkXs>)

5. React Rendering Process

- [Exploring React Rendering](<https://www.youtube.com/watch?v=i793Qm6kv3U>)

6. Virtual DOM

- [Understanding React Virtual DOM](<https://www.youtube.com/watch?v=rysTbzKOE00>)

7. React Hooks

- `useEffect`: [Mastering useEffect](<https://www.youtube.com/watch?v=UVhIMwHDS7k>)
- `useState`: [Understanding useState](<https://www.youtube.com/watch?v=kkuq0gTGRFQ>)
- `useRef`: [Demystifying useRef](<https://www.youtube.com/watch?v=yviJikU4gwk>)
- `useCallback`: [Understanding
useCallback](<https://www.youtube.com/watch?v=N62nXHTydpA>)
- `useMemo`: [Mastering useMemo](<https://www.youtube.com/watch?v=qGGu46ZoMqQ>)
- `useReducer`: [Deep Dive into
useReducer](<https://www.youtube.com/watch?v=aM1bxz-82Qw>)
- [More on useReducer](<https://www.youtube.com/watch?v=cDdGGthGA6M>)

8. Custom Hooks

- [Creating Custom React Hooks](<https://www.youtube.com/watch?v=nshyjAplovo>)
- [Mastering Custom React Hooks](<https://www.youtube.com/watch?v=NZEUDJvpQMM>)

Frontend Interview Bible

This document is broken into 9 Sections.

0. How to use this document.
1. HTML Questions
2. CSS Questions
3. Javascript Conceptual Questions
4. Javascript DOM Based Questions
5. Javascript Task Based Questions
6. Polyfills Questions
7. Machine Coding Questions
8. Frontend Design Questions
9. Performance based Questions
10. How to go beyond this document.

Philosophy behind not adding answers in the document.
Because

1. if we learn from one source, our exploration skills degrade, and our mind becomes lazy. We think that I will read it later and later never comes.
2. Answers given by might not make sense to you. Due to lack of information included which depends on the experience level of someone.
3. If we search for something to learn, our mind retains that thing for a large amount of time.

How to make most of this guide

1. Do not skip any section, Once you complete a section only move to the next section.

2. How to search answer for any question:

Step 1: Copy the question.

Step 2: Open google.com

Step 3: Paste the question in search bar

Step 4: Open first 5 Links in the new tab. (This will prepare you to foresee all possible concepts for any particular topic)

Step 5: Read from all 5 links (this is mandatory)

Step 6: After reading, write the summary anywhere (notes, online blogs wherever you like)

Step 7: Practice code examples at least 3 examples of any concepts.

HTML

1. What does H, T, M , L stand for in HTML and explain each word in detail.
2. Is HTML case sensitive or case in-sensitive?
3. Explain **<head>** tag.
4. How can I **refresh** my page every 60 seconds using HTML?
5. What are **inline** and **block** elements?
6. What is a table tag? And what is the importance of table tag.
7. What are different ways to take **input** from users?
8. How can you make an **upload file** button?
9. How can you **restrict** users to only select a .jpg file when a file upload picker is opened?
10. What does **rowspan** and **colspan** do?
11. Explain **Marquee** tag and its usage.
12. Semantic and Non Semantic Elements.
13. What is the significance of the **<noscript>** tag?
14. Describe the importance of the **<doctype>** declaration in HTML.
15. What is a picture tag and its advantages?
16. What is **srcset** attribute?
17. Explain loading=lazy and loading =eager
18. How can you change the image in the title bar?

CSS

1. What is CSS?
2. What is **inheritance** in CSS?
3. Explain **specificity** rules in CSS?
4. Explain **4 ways** to import CSS in an HTML file.
5. Can I use <style> tag in body instead of Head?
6. Explain **!important**. Can we use it !important in all CSS rules
7. How can you center a div in a webpage, horizontally and vertically. **Give 3 Possible ways.**
8. Explain **Positioning** properties.
9. What is the difference between **sticky** and **fixed**?
10. What is difference between opacity:0, visibility:hidden and display:none
11. How can I only make horizontal scroll, and disable vertical scroll.
12. Explain box model.
13. What does **calc()** function do?
14. Explain **rem** vs **em**.
15. What is the default value of rem.
16. Explain how I can change 1rem = 20px.
17. What are the benefits of using **rem**.
18. Explain **clamp** function

19. How can I show ... when some text is overflowing a container.
20. How can I **restrict** the number of lines using css.
21. How can I animate height width using css
22. Rotate a div by 45 deg
23. Explain flex, and its properties.
24. How can I change the **cursor** using CSS.
25. How can I apply CSS to the first word of a paragraph using CSS.
26. How can I check if user devices support hover or not, and based on that apply CSS.
27. Explain **sibling** selectors.
28. Explain **clearfix** property.
29. Explain how z-index works in CSS.
30. What is the maximum number the z-index supports?
31. Explain background property.
32. What are different measurement units in CSS. Give at least 6 units.
33. Explain what does :has to in CSS.
34. What are vendor prefixes in CSS.
35. How can you hide a scrollbar in CSS.
36. What CSS properties you apply on a button which is disabled.
37. What are pseudo elements?

38. How can you give a border only to the left and right of a div.
39. Difference between outline and border.
40. Explain margin padding collapsing feature.

Javascript

Conceptual Questions:

1. Explain what is Javascript?
2. Is Javascript script single threaded or multithreaded. Explain in detail.
3. How can I add Javascript in my HTML code, give me 2 possible ways.
4. Can I add a script tag in the head element?
5. Does adding script tag in head element cause any problem?
6. What are async and defer attributes?
7. Explain data types in Javascript.
8. What is hoisting? Give examples.
9. What is closure? Give Examples
10. What is stale Closure. Give Examples
11. Explain currying, it's use case and its benefits
12. What is an IIFE?
13. Give 4 ways to create an IIFE.
14. What are operators in JS.
15. How can I check data type of variables in JS.

16. Explain Abstract Equality in Javascript. By Abstract Equality I mean ==.
17. How == is different from ===
18. Explain Execution Context
19. What is Lexical Scope
20. Explain this keyword
21. What is call bind and apply?
22. Explain what is the new Keyword.
23. What are prototypes in JS?
24. Explain difference between __proto__, prototype, [[prototype]]
25. What is prototypical inheritance? How it is different from classical inheritance.
26. What is the difference between var, let and const?
27. What are Symbols in JS?
28. What is the spread and rest concept in JS.
29. Explain destructuring.
30. Explain difference between || and ??
31. What is optional chaining?
32. What is functional programming?
33. What is the difference between reduce and reduceRight.
34. What are callbacks?
35. Explain inversion of control.
36. What are promises?

37. What are the different states of promises?
38. When does the finally block get called under promises?
39. How can I handle errors using try catch in JS.
40. What is tagged template literal.
41. How does this work inside the arrow function?
42. What is the difference between an arrow and normal function? Explain 7 Differences
43. Explain Sets. How they are different from arrays.
44. What are the arguments keywords present in all functions in javascript.
45. What is “use strict” and what it does.
46. How does the for...of loop differ from the for...in loop?
47. How does the async/await syntax simplify asynchronous code in JavaScript?
48. Explain the difference between map and filter.
49. What Map and Weak Map in Javascript?
50. What are weakRef in Javascript.
51. What is difference between setTimeout and requestAnimationFrame.

DOM Manipulation and Events

1. How does event delegation work in the context of JavaScript event handling?
2. Explain the differences between the DOMContentLoaded and load events.
3. How can you stop event propagation in JavaScript?
3. Describe the purpose of the addEventListener method and how it works.
4. What is the event object, and how is it used in JavaScript event handling?
5. Explain the differences between the onclick attribute and addEventListener for handling click events.
6. Explain the concept of bubbling and capturing in event propagation.
7. How can you simulate a click event using JavaScript?
8. Explain the differences between the keyup, keydown, and keypress events.

9. What is the purpose of the preventDefault method in JavaScript events?
10. Difference between querySelector and document.getElementsByClassName

Task based FE Design Question

These questions can be first solved using plain javascript. Then using any JS Library.

In such a question the interviewer tests the skills like how do you create components, what names you use and what folder structure you follow.

1. Find the time taken by a function in Javascript.
2. Implement Infinite Scroll Using Javascript
3. How can you clear all timeouts in Javascript?
4. Print numbers from 1 - 100 with a delay of 2 sec, and also Update the numbers in UI.
5. Show a dialog, when I click a button, make sure to handle the keyboard close, and outside close of dialog.
6. Make an expanded collapse div. (Accordion Implementation). Add relevant Animations.

7. How will you implement a star rating UI?
8. Create an input text, and show suggestions as user types.
Create your own array of suggestions.
9. In The webpage, add background:yellow, to all text which are greater than 5 characters.
10. Implement a Kanban Board using Javascript.
11. You have an api, which gives some data, now you have shown all the data in the UI. Use any only hosting provider, host your own JSON and consume that JSON from your JS Code.
12. Build a calculator with basic arithmetic operations. Use only Plain JS.
13. Create a stopwatch.
14. Create a timer, once time is over show an alert.
15. Demonstrate usage of prompt and confirm box.
16. Using prompt take number from user, and based on that number print a triangle of *. Number given by the user denotes the number of rows in a triangle.
17. Move a div from left of the screen to the right of the screen.
18. Show a tooltip, on hover of a button. Make sure to give the user the ability to change tooltip position. Like left right top bottom.

19. Create a feature using which I can show notifications on my website. Like success, failure , info and warning. Just like a toast message.

Notification should be auto dismissed after 3 seconds.

There can be more than 1 notification in the screen at same time

20. Create tic-tac-toe game

21. How can you lazy load an image?

Polyfills

Polyfills are simply functions that are added on `prototype` of Constructor class, so that they can be available on all instances of the objects created by that particular constructor.

The purpose of polyfill is to support older browsers. Add new functionality in older browsers.

These days, in almost every interview you will encounter a question which demands to write a polyfill.

Below is the list of commonly asked polyfills by interviewers. There are more than 95% chances that if someone asks you to write a polyfill, that polyfill will be from the list below or must be related to the list below.

1. forEach polyfill.
2. Map polyfill.
3. Filter polyfill
4. Reduce polyfill
5. Some and Every function polyfill
6. Push and Pop polyfill
7. Slice and Splice polyfill
8. Call Bind and Apply polyfill
9. Promise.all polyfill
10. Debounce polyfill
11. Throttle polyfill
12. JSON.stringify polyfill

Machine Coding

1. You've given an array which contains some 0, your to write a function which moves all 0s to the right of the array.
2. You've an array in which each adjacent number is either +1 or -1, write a function to find the given number. Focus on efficiency.

Example: [4,5,6,7,6,5,4] // Find index of 7 (from starting)

3. Write a function to shuffle an array.
4. Write a function which gives us a fibonacci sequence of a given number.
5. You have an array, which contains some bad numbers, bad numbers are denoted with -1, write a function to remove all bad numbers from the array.
6. Write a function to flatten a given array.
7. You have a string separated by a . (dot), write a function which converts it into an object. Last character will be added as a value

Example: a.b.c.d.e output => {a : b: { c: { d: e} } }

8. Write a function which merges to given array.
9. Write a function which removes duplicates from array.
10. Write a function which converts a given function into curry version of it.

Example function sum(a,b) {} I should be able to call it like this: sum(2)(2).

11. Given an array containing n distinct numbers taken from 0, 1, 2, ..., n, find the one that is missing.
12. Write a function which deep-clones a given object.
13. Write a function which deep-compares passed value.
14. Write a function which implements ==

15. Write a function which works like a new Keyword.
16. Write a function which makes a chain of promise
17. Write a function which validates if a given set of parentheses are valid or not.
18. Implement a basic string compression algorithm. For example, the string "aaabbbcc" should be compressed to "a3b3c2".

Performance based advanced Questions

Bar Raiser Questions

1. How will you decrease the initial load time?
2. What is CDN? Explain benefits.
3. What are the different cache mechanisms provided by browsers?
4. What is http only cookie.
5. What is CORS?
6. How will you handle 10000 dom elements?
7. What is DNS lookup?
8. What is preload, prefetch and pre connect.
9. Explain SSR, ISR, and SSG
10. What are compression strategies?
11. What is DNS lookup?
12. How will you prevent XSS Attack?
13. How will you prevent CSRF Attack?
14. What is SSE and WebSocket?
15. What are different font loading strategies?
16. What are screen readers?
17. How can you make your website accessible for disabled person?
18. Explain aria attributes.

19. Explain Critical Rendering path.
20. What are 4 Web Vitals?
21. Explain package.json and package.lock JSON
22. What is Tree Shaking?
23. Explain SEO. And how can you accommodate it in SPA?
24. What is Robot.txt file?
25. Explain sitemap.xml file.
26. Explain differences between a production build and a development build.
27. What is SourceMap file.

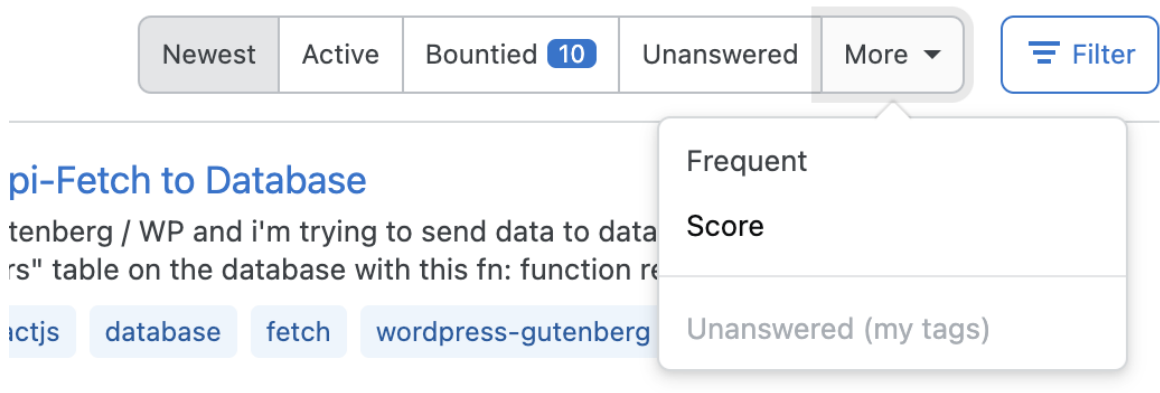
How to go beyond this document.

Tip 1

1. Go to <https://stackoverflow.com/tags>
2. Search your tag : example javascript
3. Select the tag.
4. Sort questions based on score

is NOT Java. Include all tags that are relevant to your question: e.g., [node.js], [laravel], [ember.js], [vue.js], [typescript], [svelte], etc.

1s (14)



custom context menu on textarea if word highlighted using

5. And now you have an infinite list of amazing questions which will take eternity. Now enjoy the learning.

Tip 2

Go to <https://medium.com/tag/javascript>

Now you know the drill, simply sort by newest or popularity and start learning.

JavaScript Concepts:

JavaScript:

<https://github.com/micromata/awesome-javascript-learning> (Ocean of Information)

<https://github.com/topics/javascript-resources>

<https://github.com/Asabeneh/30-Days-Of-JavaScript> - Nice to follow

Basic JavaScript Concepts:

1. Functions

- [Understanding Functions] (<https://www.youtube.com/watch?v=HkWxvB1RJq0>)
- [Deep Dive into JavaScript Functions](<https://www.youtube.com/watch?v=SHINoHxvTso>)

2. Scope in JavaScript

- [Understanding JavaScript Scope](<https://www.youtube.com/watch?v=TkFN6e9ZDMw>)
- [Mastering JavaScript Scope](<https://www.youtube.com/watch?v=uH-tVP8MUs8>)

3. Closure

- [Understanding JavaScript Closure](<https://www.youtube.com/watch?v=qikxEIxsXco>)

4. Event Loop, Call Stack, Task Queue

- [Event Loop in JavaScript](<https://www.youtube.com/watch?v=8aGhZQkoFbQ>)
- [Understanding Call Stack and Event Loop](<https://www.youtube.com/watch?v=8zKuNo4ay8E>)

5. Prototype and Prototype Chain

- [JavaScript Prototype and Inheritance](<https://www.youtube.com/watch?v=wstwjQ1yqWQ>)

6. "THIS" Keyword

- [Demystifying JavaScript "THIS"](<https://www.youtube.com/watch?v=jTPt5XO7KoU>)
- [Understanding JavaScript "THIS"](<https://www.youtube.com/watch?v=YOl79NaAtQ>)
- [In-depth Exploration of "THIS"](<https://www.youtube.com/watch?v=rv7Q11KWmKU>)

7. Class and Inheritance

- [Understanding JavaScript Classes](https://www.youtube.com/watch?v=NyCIWddeO_A)
- [JavaScript Classes and Inheritance](<https://www.youtube.com/watch?v=wstwjQ1yqWQ>)

8. DOM Manipulation

- [Manipulating the DOM with JavaScript](<https://www.youtube.com/watch?v=y17RuWkWdn8>)

9. Bind/Call/Apply

- [Mastering JavaScript Bind, Call, and Apply](<https://www.youtube.com/watch?v=75W8UPQ5I7k>)

10. Promises

- [JavaScript Promises Explained](<https://www.youtube.com/watch?v=ap-6PPAuK1Y>)
- [Deep Dive into Promises](<https://www.youtube.com/watch?v=U74BJcr8NeQ>)

11. Web APIs

- [Exploring JavaScript Web APIs](<https://www.javascripttutorial.net/web-apis/>)

12. Async/Await, Generators

- [Understanding Async/Await](<https://www.youtube.com/watch?v=SrNQS8J67zc>)

13. Functional Programming

- [Introduction to Functional Programming](https://www.youtube.com/watch?v=e-5obm1G_FY)

14. JavaScript Engine

- [Inside the JavaScript Engine](<https://www.youtube.com/watch?v=p-iiEDtpy6I>)
- [JavaScript Engine Fundamentals](<https://www.youtube.com/watch?v=2WJL19wDH68>)

15. How Browser Works

- [Understanding How the Browser Works](<https://www.youtube.com/watch?v=0IsQqJ7pwhw>)

CSS Concepts:

1. Class and Other Selectors

- [Selectors in CSS](<https://www.youtube.com/watch?v=a69OMEJXaJo>)

2. Pseudo Classes and Elements

- [Understanding CSS Pseudo Classes](<https://www.codecademy.com/resources/docs/css/pseudo-classes>)

- [CSS Pseudo Classes Cheatsheet](<https://blog.webdevsimplified.com/2021-12/css-selectors/>)

- [CSS Pseudo Elements](<https://dynamicbalaji.medium.com/css-pseudo-classes-cheatsheet-d336ee537ea0>)

4. Box Model

- [Understanding CSS Box Model](<https://www.youtube.com/watch?v=QyrYEmNYBk>)
- [Mastering CSS Box Model](<https://www.youtube.com/watch?v=rIO5326FgPE>)

5. Flex

- [CSS Flexbox](<https://www.youtube.com/watch?v=fYq5PXgSsbE>)
- [Flexbox Guide](<https://flexbox.malven.co/>)

6. Grid

- [CSS Grid](<https://www.youtube.com/watch?v=0xMQfnTU6oo>)
- [Grid Guide](<https://grid.malven.co/>)

7. Centering Elements

- [CSS Centering Techniques](<https://flaviocopes.com/css-centering/>)

8. Media Queries

- [Responsive Web Design with Media Queries](<https://www.youtube.com/watch?v=69lbzTWg5PM>)

9. CSS Preprocessors - SCSS or LESS

- [Introduction to CSS Preprocessors](<https://www.youtube.com/watch?v=69lbzTWg5PM>)

React Concepts:

1. Component State and Props

- [Understanding React State and Props](<https://www.youtube.com/watch?v=kHJSNfU7H4U>)

2. Styling Components with CSS

3. Functional and Class Components

- [Introduction to React Components](<https://www.youtube.com/watch?v=yc6elaGOoGQ>)

4. React Lifecycle Methods

- [Understanding React Lifecycle Methods](<https://www.youtube.com/watch?v=qoN5FBtfkXs>)

5. React Rendering Process

- [Exploring React Rendering](<https://www.youtube.com/watch?v=i793Qm6kv3U>)

6. Virtual DOM

- [Understanding React Virtual DOM](<https://www.youtube.com/watch?v=rysTbzKOE00>)

7. React Hooks

- `useEffect`: [Mastering useEffect](<https://www.youtube.com/watch?v=UVhIMwHDS7k>)
- `useState`: [Understanding useState](<https://www.youtube.com/watch?v=kkuq0gTGRFQ>)
- `useRef`: [Demystifying useRef](<https://www.youtube.com/watch?v=yviJikU4gwk>)
- `useCallback`: [Understanding
useCallback](<https://www.youtube.com/watch?v=N62nXHTydpA>)
- `useMemo`: [Mastering useMemo](<https://www.youtube.com/watch?v=qGGu46ZoMqQ>)
- `useReducer`: [Deep Dive into
useReducer](<https://www.youtube.com/watch?v=aM1bxz-82Qw>)
- [More on useReducer](<https://www.youtube.com/watch?v=cDdGGthGA6M>)

8. Custom Hooks

- [Creating Custom React Hooks](<https://www.youtube.com/watch?v=nshyjAplovo>)
- [Mastering Custom React Hooks](<https://www.youtube.com/watch?v=NZEUDJvpQMM>)

QUESTIONS

QUESTIONS THAT
YOU WILL BE
ASKED IN INTERVIEW,
AND HOW TO
TACKLE THEM

FRONTEND MASTER - RAHUL

QUESTIONS #1

Q Why don't you introduce your self.

A Avoid traditional intros. Introduce yourself with a story.

Start Like this,

I'm Naren. Let me share a quick story about myself. Since childhood, I've enjoyed solving puzzles, like figuring out the knots in my mother's biscuit bags. Little did I know this curiosity would lead me to my passion for debugging code and fixing bugs in programming.

After school, I joined XYZ College and started my career in 20XX as a XYZ Developer. Now, let me give you a glimpse into my typical workday and highlight some of my technical skills

QUESTIONS #2

Q Is there any application/website/work that you are proud of?

A By asking such questions, sometimes the interviewer wants to check how you explain your work, and if you know what you have written in your resume or not.

Use this question as an opportunity and explain the best work you have ever done.

Prepare for such questions before the interview, and be ready for cross questioning.

QUESTIONS #3

Q What is the biggest problem you have faced and how did you solved it?

A Answer of this questions makes it clear the scale at which you have ever worked.

If you quote a simple problem it might backfire.

So use this question as an opprtunity to showcase your skills, simply explain a extreme tough situation even tough you haven faced, but you know how to solve it.

Be prepared for cross questioning.

QUESTIONS #4

Q You came to know that your team mate has done a blunder what will you do?

A This question will check your honesty towards work.

You can simply say that you will try to help your friend to resolve that blunder before anyone knows.

And if anytime you know that you can not manage, you will escalate this and let your manager know.

QUESTIONS #5

Q You have give a problem which is above your role or skills. What will you do?

A This question will check how will you behave when put on tough spot, simply tell them I've assigned such work I will love to learn new skills considering I've give enough time to learn.

Otherwise I will take some help from my seniors, and if no help is available, and the task is critical I will spend extra time to learn and fix it.

QUESTIONS #6

Q You have applied for XYZ, Do you know what XYZ do?

A Make sure to know the basics of the company in which you are interviewing.

When HR calls its ok to say that you don't know.

But once your interview is scheduled, and the interviewer asks this, do you know what we do?

If you say NO, or even say that you only know this and this, that won't go in your favour.

Instead you can say whatever you have prepared, and ask the interviewer that you will love to hear more if you have missed anything,

QUESTIONS #7

Q Suppose you are working on a task, and you got to know another issue what will you do?

A This question checks if you are ready to take ownership or not.

Simply say if that issue is small you will fix it along with current task, and if that issue is taking time like more than 1 hour, you will report it and continue on your current task.

ALL THE BEST FOR YOUR NEXT INTERVIEW

FRONTEND MASTER - RAHUL

Frontend Interview Bible

This document is broken into 9 Sections.

0. How to use this document.
1. HTML Questions
2. CSS Questions
3. Javascript Conceptual Questions
4. Javascript DOM Based Questions
5. Javascript Task Based Questions
6. Polyfills Questions
7. Machine Coding Questions
8. Frontend Design Questions
9. Performance based Questions
10. How to go beyond this document.

Philosophy behind not adding answers in the document.
Because

1. if we learn from one source, our exploration skills degrade, and our mind becomes lazy. We think that I will read it later and later never comes.
2. Answers given by might not make sense to you. Due to lack of information included which depends on the experience level of someone.
3. If we search for something to learn, our mind retains that thing for a large amount of time.

How to make most of this guide

1. Do not skip any section, Once you complete a section only move to the next section.

2. How to search answer for any question:

Step 1: Copy the question.

Step 2: Open google.com

Step 3: Paste the question in search bar

Step 4: Open first 5 Links in the new tab. (This will prepare you to foresee all possible concepts for any particular topic)

Step 5: Read from all 5 links (this is mandatory)

Step 6: After reading, write the summary anywhere (notes, online blogs wherever you like)

Step 7: Practice code examples at least 3 examples of any concepts.

HTML

1. What does H, T, M , L stand for in HTML and explain each word in detail.
2. Is HTML case sensitive or case in-sensitive?
3. Explain **<head>** tag.
4. How can I **refresh** my page every 60 seconds using HTML?
5. What are **inline** and **block** elements?
6. What is a table tag? And what is the importance of table tag.
7. What are different ways to take **input** from users?
8. How can you make an **upload file** button?
9. How can you **restrict** users to only select a .jpg file when a file upload picker is opened?
10. What does **rowspan** and **colspan** do?
11. Explain **Marquee** tag and its usage.
12. Semantic and Non Semantic Elements.
13. What is the significance of the **<noscript>** tag?
14. Describe the importance of the **<doctype>** declaration in HTML.
15. What is a picture tag and its advantages?
16. What is **srcset** attribute?
17. Explain loading=lazy and loading =eager
18. How can you change the image in the title bar?

CSS

1. What is CSS?
2. What is **inheritance** in CSS?
3. Explain **specificity** rules in CSS?
4. Explain **4 ways** to import CSS in an HTML file.
5. Can I use <style> tag in body instead of Head?
6. Explain **!important**. Can we use it !important in all CSS rules
7. How can you center a div in a webpage, horizontally and vertically. **Give 3 Possible ways.**
8. Explain **Positioning** properties.
9. What is the difference between **sticky** and **fixed**?
10. What is difference between opacity:0, visibility:hidden and display:none
11. How can I only make horizontal scroll, and disable vertical scroll.
12. Explain box model.
13. What does **calc()** function do?
14. Explain **rem** vs **em**.
15. What is the default value of rem.
16. Explain how I can change 1rem = 20px.
17. What are the benefits of using **rem**.
18. Explain **clamp** function

19. How can I show ... when some text is overflowing a container.
20. How can I **restrict** the number of lines using css.
21. How can I animate height width using css
22. Rotate a div by 45 deg
23. Explain flex, and its properties.
24. How can I change the **cursor** using CSS.
25. How can I apply CSS to the first word of a paragraph using CSS.
26. How can I check if user devices support hover or not, and based on that apply CSS.
27. Explain **sibling** selectors.
28. Explain **clearfix** property.
29. Explain how z-index works in CSS.
30. What is the maximum number the z-index supports?
31. Explain background property.
32. What are different measurement units in CSS. Give at least 6 units.
33. Explain what does :has to in CSS.
34. What are vendor prefixes in CSS.
35. How can you hide a scrollbar in CSS.
36. What CSS properties you apply on a button which is disabled.
37. What are pseudo elements?

38. How can you give a border only to the left and right of a div.
39. Difference between outline and border.
40. Explain margin padding collapsing feature.

Javascript

Conceptual Questions:

1. Explain what is Javascript?
2. Is Javascript script single threaded or multithreaded. Explain in detail.
3. How can I add Javascript in my HTML code, give me 2 possible ways.
4. Can I add a script tag in the head element?
5. Does adding script tag in head element cause any problem?
6. What are async and defer attributes?
7. Explain data types in Javascript.
8. What is hoisting? Give examples.
9. What is closure? Give Examples
10. What is stale Closure. Give Examples
11. Explain currying, it's use case and its benefits
12. What is an IIFE?
13. Give 4 ways to create an IIFE.
14. What are operators in JS.
15. How can I check data type of variables in JS.

16. Explain Abstract Equality in Javascript. By Abstract Equality I mean ==.
17. How == is different from ===
18. Explain Execution Context
19. What is Lexical Scope
20. Explain this keyword
21. What is call bind and apply?
22. Explain what is the new Keyword.
23. What are prototypes in JS?
24. Explain difference between __proto__, prototype, [[prototype]]
25. What is prototypical inheritance? How it is different from classical inheritance.
26. What is the difference between var, let and const?
27. What are Symbols in JS?
28. What is the spread and rest concept in JS.
29. Explain destructuring.
30. Explain difference between || and ??
31. What is optional chaining?
32. What is functional programming?
33. What is the difference between reduce and reduceRight.
34. What are callbacks?
35. Explain inversion of control.
36. What are promises?

37. What are the different states of promises?
38. When does the finally block get called under promises?
39. How can I handle errors using try catch in JS.
40. What is tagged template literal.
41. How does this work inside the arrow function?
42. What is the difference between an arrow and normal function? Explain 7 Differences
43. Explain Sets. How they are different from arrays.
44. What are the arguments keywords present in all functions in javascript.
45. What is “use strict” and what it does.
46. How does the for...of loop differ from the for...in loop?
47. How does the async/await syntax simplify asynchronous code in JavaScript?
48. Explain the difference between map and filter.
49. What Map and Weak Map in Javascript?
50. What are weakRef in Javascript.
51. What is difference between setTimeout and requestAnimationFrame.

DOM Manipulation and Events

1. How does event delegation work in the context of JavaScript event handling?
2. Explain the differences between the DOMContentLoaded and load events.
3. How can you stop event propagation in JavaScript?
3. Describe the purpose of the addEventListener method and how it works.
4. What is the event object, and how is it used in JavaScript event handling?
5. Explain the differences between the onclick attribute and addEventListener for handling click events.
6. Explain the concept of bubbling and capturing in event propagation.
7. How can you simulate a click event using JavaScript?
8. Explain the differences between the keyup, keydown, and keypress events.

9. What is the purpose of the preventDefault method in JavaScript events?
10. Difference between querySelector and document.getElementsByClassName

Task based FE Design Question

These questions can be first solved using plain javascript. Then using any JS Library.

In such a question the interviewer tests the skills like how do you create components, what names you use and what folder structure you follow.

1. Find the time taken by a function in Javascript.
2. Implement Infinite Scroll Using Javascript
3. How can you clear all timeouts in Javascript?
4. Print numbers from 1 - 100 with a delay of 2 sec, and also Update the numbers in UI.
5. Show a dialog, when I click a button, make sure to handle the keyboard close, and outside close of dialog.
6. Make an expanded collapse div. (Accordion Implementation). Add relevant Animations.

7. How will you implement a star rating UI?
8. Create an input text, and show suggestions as user types.
Create your own array of suggestions.
9. In The webpage, add background:yellow, to all text which
are greater than 5 characters.
10. Implement a Kanban Board using Javascript.
11. You have an api, which gives some data, now you have
shown all the data in the UI. Use any only hosting provider,
host your own JSON and consume that JSON from your JS
Code.
12. Build a calculator with basic arithmetic operations. Use
only Plain JS.
13. Create a stopwatch.
14. Create a timer, once time is over show an alert.
15. Demonstrate usage of prompt and confirm box.
16. Using prompt take number from user, and based on that
number print a triangle of *. Number given by the user
denotes the number of rows in a triangle.
17. Move a div from left of the screen to the right of the
screen.
18. Show a tooltip, on hover of a button. Make sure to give the
user the ability to change tooltip position. Like left right top
bottom.

19. Create a feature using which I can show notifications on my website. Like success, failure , info and warning. Just like a toast message.

Notification should be auto dismissed after 3 seconds.

There can be more than 1 notification in the screen at same time

20. Create tic-tac-toe game

21. How can you lazy load an image?

Polyfills

Polyfills are simply functions that are added on `prototype` of Constructor class, so that they can be available on all instances of the objects created by that particular constructor.

The purpose of polyfill is to support older browsers. Add new functionality in older browsers.

These days, in almost every interview you will encounter a question which demands to write a polyfill.

Below is the list of commonly asked polyfills by interviewers. There are more than 95% chances that if someone asks you to write a polyfill, that polyfill will be from the list below or must be related to the list below.

1. forEach polyfill.
2. Map polyfill.
3. Filter polyfill
4. Reduce polyfill
5. Some and Every function polyfill
6. Push and Pop polyfill
7. Slice and Splice polyfill
8. Call Bind and Apply polyfill
9. Promise.all polyfill
10. Debounce polyfill
11. Throttle polyfill
12. JSON.stringify polyfill

Machine Coding

1. You've given an array which contains some 0, your to write a function which moves all 0s to the right of the array.
2. You've an array in which each adjacent number is either +1 or -1, write a function to find the given number. Focus on efficiency.

Example: [4,5,6,7,6,5,4] // Find index of 7 (from starting)

3. Write a function to shuffle an array.
4. Write a function which gives us a fibonacci sequence of a given number.
5. You have an array, which contains some bad numbers, bad numbers are denoted with -1, write a function to remove all bad numbers from the array.
6. Write a function to flatten a given array.
7. You have a string separated by a . (dot), write a function which converts it into an object. Last character will be added as a value

Example: a.b.c.d.e output => {a : b: { c: { d: e} } }

8. Write a function which merges to given array.
9. Write a function which removes duplicates from array.
10. Write a function which converts a given function into curry version of it.

Example function sum(a,b) {} I should be able to call it like this: sum(2)(2).

11. Given an array containing n distinct numbers taken from 0, 1, 2, ..., n, find the one that is missing.
12. Write a function which deep-clones a given object.
13. Write a function which deep-compares passed value.
14. Write a function which implements ==

15. Write a function which works like a new Keyword.
16. Write a function which makes a chain of promise
17. Write a function which validates if a given set of parentheses are valid or not.
18. Implement a basic string compression algorithm. For example, the string "aaabbbcc" should be compressed to "a3b3c2".

Performance based advanced Questions

Bar Raiser Questions

1. How will you decrease the initial load time?
2. What is CDN? Explain benefits.
3. What are the different cache mechanisms provided by browsers?
4. What is http only cookie.
5. What is CORS?
6. How will you handle 10000 dom elements?
7. What is DNS lookup?
8. What is preload, prefetch and pre connect.
9. Explain SSR, ISR, and SSG
10. What are compression strategies?
11. What is DNS lookup?
12. How will you prevent XSS Attack?
13. How will you prevent CSRF Attack?
14. What is SSE and WebSocket?
15. What are different font loading strategies?
16. What are screen readers?
17. How can you make your website accessible for disabled person?
18. Explain aria attributes.

19. Explain Critical Rendering path.
20. What are 4 Web Vitals?
21. Explain package.json and package.lock JSON
22. What is Tree Shaking?
23. Explain SEO. And how can you accommodate it in SPA?
24. What is Robot.txt file?
25. Explain sitemap.xml file.
26. Explain differences between a production build and a development build.
27. What is SourceMap file.

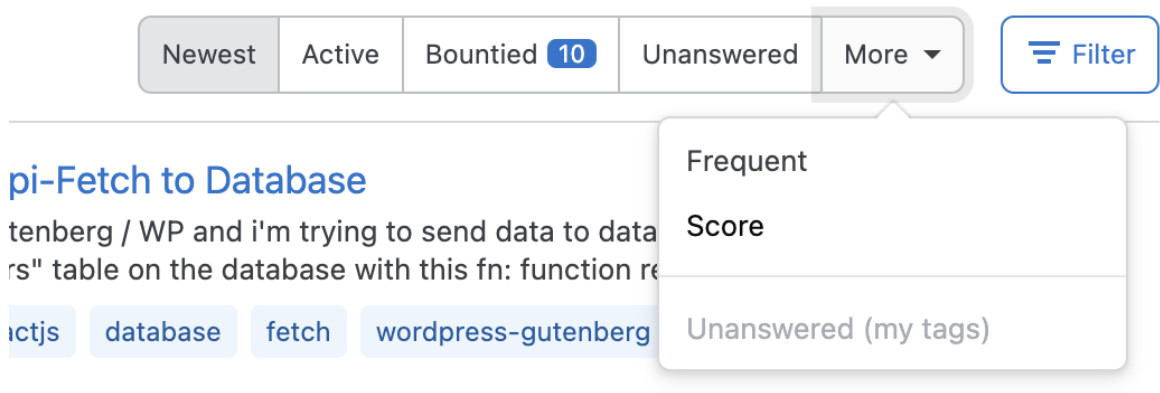
How to go beyond this document.

Tip 1

1. Go to <https://stackoverflow.com/tags>
2. Search your tag : example javascript
3. Select the tag.
4. Sort questions based on score

is NOT Java. Include all tags that are relevant to your question: e.g., [node.js], [laravel], [ember.js], [vue.js], [typescript], [svelte], etc.

1s (14)



custom context menu on textarea if word highlighted using

5. And now you have an infinite list of amazing questions which will take eternity. Now enjoy the learning.

Tip 2

Go to <https://medium.com/tag/javascript>

Now you know the drill, simply sort by newest or popularity and start learning.

QUESTIONS

QUESTIONS THAT
YOU WILL BE
ASKED IN INTERVIEW,
AND HOW TO
TACKLE THEM

FRONTEND MASTER - RAHUL

QUESTIONS #1

Q Why don't you introduce your self.

A Avoid traditional intros. Introduce yourself with a story.

Start Like this,

I'm Naren. Let me share a quick story about myself. Since childhood, I've enjoyed solving puzzles, like figuring out the knots in my mother's biscuit bags. Little did I know this curiosity would lead me to my passion for debugging code and fixing bugs in programming.

After school, I joined XYZ College and started my career in 20XX as a XYZ Developer. Now, let me give you a glimpse into my typical workday and highlight some of my technical skills

QUESTIONS #2

Q Is there any application/website/work that you are proud of?

A By asking such questions, sometimes interviewers want to check how you explain your work, and if you know what you have written in your resume or not.

Use this question as an opportunity and explain the best work you have ever done.

Prepare for such questions before interview, and be ready for cross questioning.

QUESTIONS #3

Q What is the biggest problem you have faced and how did you solved it?

A Answer of this questions makes it clear the scale at which you have ever worked.

If you quote a simple problem it might backfire.

So use this question as an oppprtunity to showcase your skills, simply explain a extreme tough situation even tough you haven faced, but you know how to solve it.

Be prepared for cross questioning.

QUESTIONS #4

Q You came to know that your team mate has done a blunder what will you do?

A This question will check your honesty towards work.

You can simply say that you will try to help your friend to resolve that blunder before anyone knows.

And if anytime you know that you can not manage, you will escalate this and let your manager know.

QUESTIONS #5

Q You have give a problem which is above your role or skills. What will you do?

A This question will check how will you behave when put on tough spot, simply tell them I've assigned such work I will love to learn new skills considering I've give enough time to learn.

Otherwise I will take some help from my seniors, and if no help is available, and the task is critical I will spend extra time to learn and fix it.

QUESTIONS #6

Q You have applied for XYZ, Do you know what XYZ do?

A Make sure to know the basics of the company in which you are interviewing.

When HR calls its ok to say that you don't know.

But once your interview is scheduled, and the interviewer asks this, do you know what we do?

If you say NO, or even say that you only know this and this, that won't go in your favour.

Instead you can say whatever you have prepared, and ask the interviewer that you will love to hear more if you have missed anything,

QUESTIONS #7

Q Suppose you are working on a task, and you got to know another issue what will you do?

A This question checks if you are ready to take ownership or not.

Simply say if that issue is small you will fix it along with current task, and if that issue is taking time like more than 1 hour, you will report it and continue on your current task.

ALL THE BEST FOR YOUR NEXT INTERVIEW

FRONTEND MASTER - RAHUL

Frontend Awareness Must Known Things

1. Explain What is DNS? And How it Works?
2. Explain the Difference Between Authentication and Authorization.
3. What is the Purpose of a Session Token in Web Security?
4. Explain the Role of JSON Web Tokens (JWT) in Authentication.
5. What is TCP?
6. Describe the Three-Way Handshake in TCP.
7. What is SSL?
8. How client server works?
9. How will you deploy your application to server?
10. When you press enter in addressbar. Explain whole process from pressing enter to showing content on browser.
11. What is SEO?
12. What is difference between SPA and traditional Web apps.
13. What is an API?
14. OOPS Concepts.
15. Race Conditions, Deadlocks.
16. Parallel vs Concurrency in programming
17. What do you understand by CMS.
18. What is HTTP / HTTP 2.
19. Explain REST.
20. How does Router works.
21. Explain LAN MAN and WAN
22. Web Security - How to you prevent security attacks
23. Explain the Role of a Firewall in Network Security.
24. What is CI/CD
25. Git and GitHub
26. Exposure to any design tool like Figma (Not the designing but should be aware about how to use)
27. What is pixel?

Frontend Interview Bible

This document is broken into 9 Sections.

0. How to use this document.
1. HTML Questions
2. CSS Questions
3. Javascript Conceptual Questions
4. Javascript DOM Based Questions
5. Javascript Task Based Questions
6. Polyfills Questions
7. Machine Coding Questions
8. Frontend Design Questions
9. Performance based Questions
10. How to go beyond this document.

Philosophy behind not adding answers in the document.
Because

1. if we learn from one source, our exploration skills degrade, and our mind becomes lazy. We think that I will read it later and later never comes.
2. Answers given by might not make sense to you. Due to lack of information included which depends on the experience level of someone.
3. If we search for something to learn, our mind retains that thing for a large amount of time.

How to make most of this guide

1. Do not skip any section, Once you complete a section only move to the next section.

2. How to search answer for any question:

Step 1: Copy the question.

Step 2: Open google.com

Step 3: Paste the question in search bar

Step 4: Open first 5 Links in the new tab. (This will prepare you to foresee all possible concepts for any particular topic)

Step 5: Read from all 5 links (this is mandatory)

Step 6: After reading, write the summary anywhere (notes, online blogs wherever you like)

Step 7: Practice code examples at least 3 examples of any concepts.

HTML

1. What does H, T, M , L stand for in HTML and explain each word in detail.
2. Is HTML case sensitive or case in-sensitive?
3. Explain **<head>** tag.
4. How can I **refresh** my page every 60 seconds using HTML?
5. What are **inline** and **block** elements?
6. What is a table tag? And what is the importance of table tag.
7. What are different ways to take **input** from users?
8. How can you make an **upload file** button?
9. How can you **restrict** users to only select a .jpg file when a file upload picker is opened?
10. What does **rowspan** and **colspan** do?
11. Explain **Marquee** tag and its usage.
12. Semantic and Non Semantic Elements.
13. What is the significance of the **<noscript>** tag?
14. Describe the importance of the **<doctype>** declaration in HTML.
15. What is a picture tag and its advantages?
16. What is **srcset** attribute?
17. Explain loading=lazy and loading =eager
18. How can you change the image in the title bar?

CSS

1. What is CSS?
2. What is **inheritance** in CSS?
3. Explain **specificity** rules in CSS?
4. Explain **4 ways** to import CSS in an HTML file.
5. Can I use <style> tag in body instead of Head?
6. Explain **!important**. Can we use it !important in all CSS rules
7. How can you center a div in a webpage, horizontally and vertically. **Give 3 Possible ways.**
8. Explain **Positioning** properties.
9. What is the difference between **sticky** and **fixed**?
10. What is difference between opacity:0, visibility:hidden and display:none
11. How can I only make horizontal scroll, and disable vertical scroll.
12. Explain box model.
13. What does **calc()** function do?
14. Explain **rem** vs **em**.
15. What is the default value of rem.
16. Explain how I can change 1rem = 20px.
17. What are the benefits of using **rem**.
18. Explain **clamp** function

19. How can I show ... when some text is overflowing a container.
20. How can I **restrict** the number of lines using css.
21. How can I animate height width using css
22. Rotate a div by 45 deg
23. Explain flex, and its properties.
24. How can I change the **cursor** using CSS.
25. How can I apply CSS to the first word of a paragraph using CSS.
26. How can I check if user devices support hover or not, and based on that apply CSS.
27. Explain **sibling** selectors.
28. Explain **clearfix** property.
29. Explain how z-index works in CSS.
30. What is the maximum number the z-index supports?
31. Explain background property.
32. What are different measurement units in CSS. Give at least 6 units.
33. Explain what does :has to in CSS.
34. What are vendor prefixes in CSS.
35. How can you hide a scrollbar in CSS.
36. What CSS properties you apply on a button which is disabled.
37. What are pseudo elements?

38. How can you give a border only to the left and right of a div.
39. Difference between outline and border.
40. Explain margin padding collapsing feature.

Javascript

Conceptual Questions:

1. Explain what is Javascript?
2. Is Javascript script single threaded or multithreaded. Explain in detail.
3. How can I add Javascript in my HTML code, give me 2 possible ways.
4. Can I add a script tag in the head element?
5. Does adding script tag in head element cause any problem?
6. What are async and defer attributes?
7. Explain data types in Javascript.
8. What is hoisting? Give examples.
9. What is closure? Give Examples
10. What is stale Closure. Give Examples
11. Explain currying, it's use case and its benefits
12. What is an IIFE?
13. Give 4 ways to create an IIFE.
14. What are operators in JS.
15. How can I check data type of variables in JS.

16. Explain Abstract Equality in Javascript. By Abstract Equality I mean ==.
17. How == is different from ===
18. Explain Execution Context
19. What is Lexical Scope
20. Explain this keyword
21. What is call bind and apply?
22. Explain what is the new Keyword.
23. What are prototypes in JS?
24. Explain difference between __proto__, prototype, [[prototype]]
25. What is prototypical inheritance? How it is different from classical inheritance.
26. What is the difference between var, let and const?
27. What are Symbols in JS?
28. What is the spread and rest concept in JS.
29. Explain destructuring.
30. Explain difference between || and ??
31. What is optional chaining?
32. What is functional programming?
33. What is the difference between reduce and reduceRight.
34. What are callbacks?
35. Explain inversion of control.
36. What are promises?

37. What are the different states of promises?
38. When does the finally block get called under promises?
39. How can I handle errors using try catch in JS.
40. What is tagged template literal.
41. How does this work inside the arrow function?
42. What is the difference between an arrow and normal function? Explain 7 Differences
43. Explain Sets. How they are different from arrays.
44. What are the arguments keywords present in all functions in javascript.
45. What is “use strict” and what it does.
46. How does the for...of loop differ from the for...in loop?
47. How does the async/await syntax simplify asynchronous code in JavaScript?
48. Explain the difference between map and filter.
49. What Map and Weak Map in Javascript?
50. What are weakRef in Javascript.
51. What is difference between setTimeout and requestAnimationFrame.

DOM Manipulation and Events

1. How does event delegation work in the context of JavaScript event handling?
2. Explain the differences between the DOMContentLoaded and load events.
3. How can you stop event propagation in JavaScript?
3. Describe the purpose of the addEventListener method and how it works.
4. What is the event object, and how is it used in JavaScript event handling?
5. Explain the differences between the onclick attribute and addEventListener for handling click events.
6. Explain the concept of bubbling and capturing in event propagation.
7. How can you simulate a click event using JavaScript?
8. Explain the differences between the keyup, keydown, and keypress events.

9. What is the purpose of the preventDefault method in JavaScript events?
10. Difference between querySelector and document.getElementsByClassName

Task based FE Design Question

These questions can be first solved using plain javascript. Then using any JS Library.

In such a question the interviewer tests the skills like how do you create components, what names you use and what folder structure you follow.

1. Find the time taken by a function in Javascript.
2. Implement Infinite Scroll Using Javascript
3. How can you clear all timeouts in Javascript?
4. Print numbers from 1 - 100 with a delay of 2 sec, and also Update the numbers in UI.
5. Show a dialog, when I click a button, make sure to handle the keyboard close, and outside close of dialog.
6. Make an expanded collapse div. (Accordion Implementation). Add relevant Animations.

7. How will you implement a star rating UI?
8. Create an input text, and show suggestions as user types.
Create your own array of suggestions.
9. In The webpage, add background:yellow, to all text which
are greater than 5 characters.
10. Implement a Kanban Board using Javascript.
11. You have an api, which gives some data, now you have
shown all the data in the UI. Use any only hosting provider,
host your own JSON and consume that JSON from your JS
Code.
12. Build a calculator with basic arithmetic operations. Use
only Plain JS.
13. Create a stopwatch.
14. Create a timer, once time is over show an alert.
15. Demonstrate usage of prompt and confirm box.
16. Using prompt take number from user, and based on that
number print a triangle of *. Number given by the user
denotes the number of rows in a triangle.
17. Move a div from left of the screen to the right of the
screen.
18. Show a tooltip, on hover of a button. Make sure to give the
user the ability to change tooltip position. Like left right top
bottom.

19. Create a feature using which I can show notifications on my website. Like success, failure , info and warning. Just like a toast message.

Notification should be auto dismissed after 3 seconds.

There can be more than 1 notification in the screen at same time

20. Create tic-tac-toe game

21. How can you lazy load an image?

Polyfills

Polyfills are simply functions that are added on `prototype` of Constructor class, so that they can be available on all instances of the objects created by that particular constructor.

The purpose of polyfill is to support older browsers. Add new functionality in older browsers.

These days, in almost every interview you will encounter a question which demands to write a polyfill.

Below is the list of commonly asked polyfills by interviewers. There are more than 95% chances that if someone asks you to write a polyfill, that polyfill will be from the list below or must be related to the list below.

1. forEach polyfill.
2. Map polyfill.
3. Filter polyfill
4. Reduce polyfill
5. Some and Every function polyfill
6. Push and Pop polyfill
7. Slice and Splice polyfill
8. Call Bind and Apply polyfill
9. Promise.all polyfill
10. Debounce polyfill
11. Throttle polyfill
12. JSON.stringify polyfill

Machine Coding

1. You've given an array which contains some 0, your to write a function which moves all 0s to the right of the array.
2. You've an array in which each adjacent number is either +1 or -1, write a function to find the given number. Focus on efficiency.

Example: [4,5,6,7,6,5,4] // Find index of 7 (from starting)

3. Write a function to shuffle an array.
4. Write a function which gives us a fibonacci sequence of a given number.
5. You have an array, which contains some bad numbers, bad numbers are denoted with -1, write a function to remove all bad numbers from the array.
6. Write a function to flatten a given array.
7. You have a string separated by a . (dot), write a function which converts it into an object. Last character will be added as a value

Example: a.b.c.d.e output => {a : b: { c: { d: e} } }

8. Write a function which merges to given array.
9. Write a function which removes duplicates from array.
10. Write a function which converts a given function into curry version of it.

Example function sum(a,b) {} I should be able to call it like this: sum(2)(2).

11. Given an array containing n distinct numbers taken from 0, 1, 2, ..., n, find the one that is missing.
12. Write a function which deep-clones a given object.
13. Write a function which deep-compares passed value.
14. Write a function which implements ==

15. Write a function which works like a new Keyword.
16. Write a function which makes a chain of promise
17. Write a function which validates if a given set of parentheses are valid or not.
18. Implement a basic string compression algorithm. For example, the string "aaabbbcc" should be compressed to "a3b3c2".

Performance based advanced Questions

Bar Raiser Questions

1. How will you decrease the initial load time?
2. What is CDN? Explain benefits.
3. What are the different cache mechanisms provided by browsers?
4. What is http only cookie.
5. What is CORS?
6. How will you handle 10000 dom elements?
7. What is DNS lookup?
8. What is preload, prefetch and pre connect.
9. Explain SSR, ISR, and SSG
10. What are compression strategies?
11. What is DNS lookup?
12. How will you prevent XSS Attack?
13. How will you prevent CSRF Attack?
14. What is SSE and WebSocket?
15. What are different font loading strategies?
16. What are screen readers?
17. How can you make your website accessible for disabled person?
18. Explain aria attributes.

19. Explain Critical Rendering path.
20. What are 4 Web Vitals?
21. Explain package.json and package.lock JSON
22. What is Tree Shaking?
23. Explain SEO. And how can you accommodate it in SPA?
24. What is Robot.txt file?
25. Explain sitemap.xml file.
26. Explain differences between a production build and a development build.
27. What is SourceMap file.

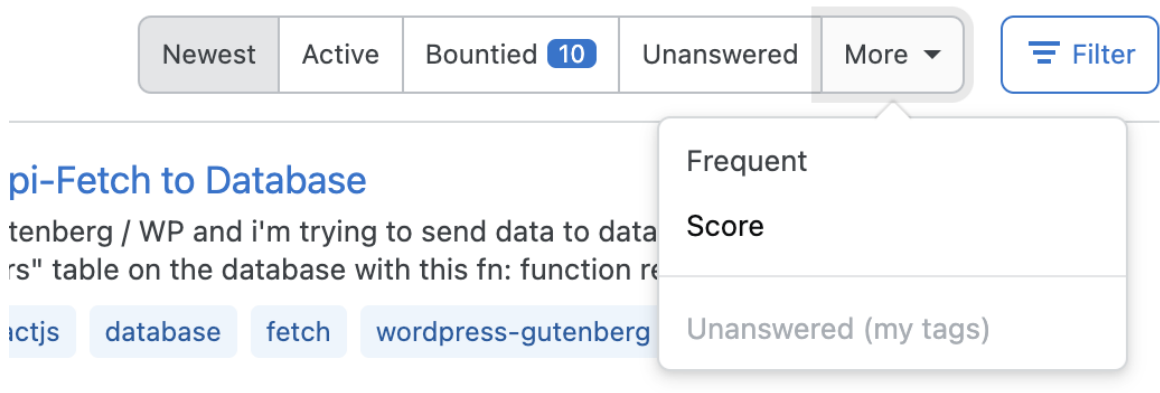
How to go beyond this document.

Tip 1

1. Go to <https://stackoverflow.com/tags>
2. Search your tag : example javascript
3. Select the tag.
4. Sort questions based on score

is NOT Java. Include all tags that are relevant to your question: e.g., [node.js], [laravel], [ember.js], [vue.js], [typescript], [svelte], etc.

1s (14)



custom context menu on textarea if word highlighted using

5. And now you have an infinite list of amazing questions which will take eternity. Now enjoy the learning.

Tip 2

Go to <https://medium.com/tag/javascript>

Now you know the drill, simply sort by newest or popularity and start learning.

QUESTIONS

QUESTIONS THAT
YOU WILL BE
ASKED IN INTERVIEW,
AND HOW TO
TACKLE THEM

FRONTEND MASTER - RAHUL

QUESTIONS #1

Q Why don't you introduce your self.

A Avoid traditional intros. Introduce yourself with a story.

Start Like this,

I'm Naren. Let me share a quick story about myself. Since childhood, I've enjoyed solving puzzles, like figuring out the knots in my mother's biscuit bags. Little did I know this curiosity would lead me to my passion for debugging code and fixing bugs in programming.

After school, I joined XYZ College and started my career in 20XX as a XYZ Developer. Now, let me give you a glimpse into my typical workday and highlight some of my technical skills

QUESTIONS #2

Q Is there any application/website/work that you are proud of?

A By asking such questions, sometimes interviewers want to check how you explain your work, and if you know what you have written in your resume or not.

Use this question as an opportunity and explain the best work you have ever done.

Prepare for such questions before interview, and be ready for cross questioning.

QUESTIONS #3

Q What is the biggest problem you have faced and how did you solved it?

A Answer of this questions makes it clear the scale at which you have ever worked.

If you quote a simple problem it might backfire.

So use this question as an oppprtunity to showcase your skills, simply explain a extreme tough situation even tough you haven faced, but you know how to solve it.

Be prepared for cross questioning.

QUESTIONS #4

Q You came to know that your team mate has done a blunder what will you do?

A This question will check your honesty towards work.

You can simply say that you will try to help your friend to resolve that blunder before anyone knows.

And if anytime you know that you can not manage, you will escalate this and let your manager know.

QUESTIONS #5

Q You have give a problem which is above your role or skills. What will you do?

A This question will check how will you behave when put on tough spot, simply tell them I've assigned such work I will love to learn new skills considering I've give enough time to learn.

Otherwise I will take some help from my seniors, and if no help is available, and the task is critical I will spend extra time to learn and fix it.

QUESTIONS #6

Q You have applied for XYZ, Do you know what XYZ do?

A Make sure to know the basics of the company in which you are interviewing.

When HR calls its ok to say that you don't know.

But once your interview is scheduled, and the interviewer asks this, do you know what we do?

If you say NO, or even say that you only know this and this, that won't go in your favour.

Instead you can say whatever you have prepared, and ask the interviewer that you will love to hear more if you have missed anything,

QUESTIONS #7

Q Suppose you are working on a task, and you got to know another issue what will you do?

A This question checks if you are ready to take ownership or not.

Simply say if that issue is small you will fix it along with current task, and if that issue is taking time like more than 1 hour, you will report it and continue on your current task.

ALL THE BEST FOR YOUR NEXT INTERVIEW

FRONTEND MASTER - RAHUL

Frontend Awareness Must Known Things

1. Explain What is DNS? And How it Works?
2. Explain the Difference Between Authentication and Authorization.
3. What is the Purpose of a Session Token in Web Security?
4. Explain the Role of JSON Web Tokens (JWT) in Authentication.
5. What is TCP?
6. Describe the Three-Way Handshake in TCP.
7. What is SSL?
8. How client server works?
9. How will you deploy your application to server?
10. When you press enter in addressbar. Explain whole process from pressing enter to showing content on browser.
11. What is SEO?
12. What is difference between SPA and traditional Web apps.
13. What is an API?
14. OOPS Concepts.
15. Race Conditions, Deadlocks.
16. Parallel vs Concurrency in programming
17. What do you understand by CMS.
18. What is HTTP / HTTP 2.
19. Explain REST.
20. How does Router works.
21. Explain LAN MAN and WAN
22. Web Security - How to you prevent security attacks
23. Explain the Role of a Firewall in Network Security.
24. What is CI/CD
25. Git and GitHub
26. Exposure to any design tool like Figma (Not the designing but should be aware about how to use)
27. What is pixel?

