General Questions:

1. How do We optimize a website's assets/resources?
2. What are the ways to reduce page load time?
3. What kind of things must We be wary of when design or developing for multilingual sites?
4. Can We describeWe r workflow when We create a web page?
5. Can We describe the difference between progressive enhancement and graceful degradation?
6. How many resources will a browser download from a given domain at a time?
    a. What are the exceptions?
7. Explain what ARIA and screenreaders are, and how to make a website accessible.
8. What does CORS stand for and what issue does it address?
9. How to handle cross-origin issues ?
10. If We Have An Issue With We r Page, How Do We Debug It, What Tools Do We Use?Ans- debugging practices like W3c validator, Firebug, Chrome Dev Tools.

11. How to avoid cross-browser compatibility issues.

HTML Questions:

1. What does a **!`DOCTYPE`** do?
2. What happens when we don't write DOCTYPE?
3. DOCTYPE is case sensitive?
4. What is the use of data- attribute?
5. What all tags are important in head section of html file?
6. What is the use of <meta> tag?
7. What's the difference between full standards mode and quirks mode?
8. How do We serve a page with content in multiple languages?
9. What kind of things must We beaware of when design or developing for multilingual sites?
10. What are `data-` attributes good for?
11. What are the building blocks of HTML5?
12. Describe the difference between a `cookie`, `sessionStorage` and `localStorage`.
13. Describe the difference between `<script>`, `<script async>` and `<script defer>`.
14. Why is it generally a good idea to position CSS `<link>`s between `<head></head>` and JS `<script>`s just before `</body>`? Do We know any exceptions?
15. What is progressive rendering?
16. Why We would use a `srcset` attribute in an image tag? Explain the process the browser uses when evaluating the content of this attribute.
17. Have We used different HTML templating languages before?
18. Div vs Span
19. What is the difference between form get and form post?
20. SVG vs Canvas
21. LocalStoragevs session storage?
22. What is the lifetime of local storage?
23. What is the difference between local storage and cookies?

24. What are some of the key new features in HTML5?

CSS Questions:

1. What is the difference between classes and IDs in CSS?
2. What's the difference between "resetting" and "normalizing" CSS? Which would We choose, and why?
3. Describe Floats and how they work.
4. Describe z-index and how stacking context is formed.
5. Describe BFC(Block Formatting Context) and how it works.
6. What are the various clearing techniques and which is appropriate for what context?
7. Explain CSS sprites, and how We would implement them on a page or site.
8. What are We r favourite image replacement techniques and which do We use when?
9. How would We approach fixing browser-specific styling issues?
10. How do We serveWe r pages for feature-constrained browsers?
    - o What techniques/processes do We use?
11. What are the different ways to visually hide content (and make it available only for screen readers)?
12. Have We ever used a grid system, and if so, what do We prefer?
13. Have We used or implemented media queries or mobile specific laWets/CSS?
14. Are We familiar with styling SVG?
15. How do We optimizeWe r webpages for print?
16. What are some of the "gotchas" for writing efficient CSS?
17. What are the advantages/disadvantages of using CSS preprocessors?
    - o Describe what We like and dislike about the CSS preprocessors We have used.
18. How would We implement a web design comp that uses non-standard fonts?
19. Explain how a browser determines what elements match a CSS selector.
20. Describe pseudo-elements and discuss what they are used for.
21. Explain We r understanding of the box model and how We would tell the browser in CSS to render We r laWet in different box models.
22. What does `* { box-sizing: border-box; }` do? What are its advantages?
23. List as many values for the display property that We can remember.
24. What's the difference between inline and inline-block?
25. What's the difference between a relative, fixed, absolute and statically positioned element?
26. The 'C' in CSS stands for Cascading. How is priority determined in assigning styles (a few examples)? How can We use this system to We r advantage?
27. What existing CSS frameworks have We used locally, or in production? How would We change/improve them?
28. Have We played around with the new CSS Flexbox or Grid specs?
29. How is responsive design different from adaptive design?
30. Have We ever worked with retina graphics? If so, when and what techniques did We use?
31. Is there any reason We 'd want to use `translate()` instead of *absolute positioning*, or vice-versa? And why?
32. What is the difference between inline, inline-block, and block?
33. What is the "Box Model" in CSS? Which CSS properties are a part of it?

34. Explain the difference between visibility:hidden; and display:none; ?
35. How do We clear a floated element?
36. Pseudo elements vs pseudo classes
37. How to align a div center both vertically and Horizontally.
38. What is a sprite? How is it applied using CSS? What is the benefit?
39. What are some accessibility concerns that come up in CSS?
40. What Is A CssReset. What Is The Difference Between A Css Reset And Normalize.css
41. Ans-CSS Reset removes browser default styles. Normalize.css sets a standard across all browsers (It does not 'reset' them)
42. How Would We Solve A Floated Div's Parent Height.
43. Ans-Clearfix, Float Parent as well, User overflow property other than 'visible'.
44. What tools do We use for cross-browser testing?
45. They should have some kind of strategy. Perhaps a web-based tool like BrowserStack. Perhaps a VM based tool like Virtual Box. Perhaps different actual computers. Part of the job of front end design is making sure things work everywhere they can (based on decided-upon support). We don't have to love it, but We can't hate it. "This right here, this is the job. What kind of work were We expecting?

JS Questions:

1. Explain event delegation
2. Explain how `this` works in JavaScript
3. Explain how prototypal inheritance works
4. Explain why the following doesn't work as an IIFE: `function foo(){ }();`.
   - o What needs to be changed to properly make it an IIFE?
5. What's the difference between a variable that is: `null`, `undefined` or undeclared?
   - o How would We go about checking for any of these states?
6. What is a closure, and how/why would We use one?
7. Can We describe the main difference between a `forEach` loop and a `.map()` loop and why We would pick one versus the other?
8. What's a typical use case for anonymous functions?
9. How do We organizeWe r code? (module pattern, classical inheritance?)
10. Explain the difference between classical inheritance and prototypal inheritance.
11. What's the difference between host objects and native objects?
12. Difference between: `function Person(){}`, `var person = Person()`, and `var person = new Person()`?
13. What's the difference between `.call` and `.apply`?
14. Explain `Function.prototype.bind`.
15. When would We use `document.write()`?
16. What's the difference between feature detection, feature inference, and using the UA string?
17. Explain Ajax in as much detail as possible.
18. What are the advantages and disadvantages of using Ajax?
19. Explain how JSONP works (and how it's not really Ajax).
20. Have We ever used JavaScript templating?
   - o If so, what libraries have We used?
21. Explain "hoisting".
22. Describe event bubbling.

23. What's the difference between an "attribute" and a "property"?
24. Why is extending built-in JavaScript objects not a good idea?
25. Difference between document load event and document DOMContentLoaded event?
26. What is the difference between == and ===?
27. Explain the same-origin policy with regards to JavaScript.
28. Make this work:
    - `duplicate([1,2,3,4,5]); // [1,2,3,4,5,1,2,3,4,5]`
29. Why is it called a Ternary expression, what does the word "Ternary" indicate?
30. What is `"use strict";`?what are the advantages and disadvantages to using it?
31. Create a for loop that iterates up to `100` while outputting **"fizz"** at multiples of `3`, **"buzz"** at multiples of `5` and **"fizzbuzz"** at multiples of `3` and `5`
32. Why is it, in general, a good idea to leave the global scope of a website as-is and never touch it?
33. Why would We use something like the `load` event? Does this event have disadvantages? Do We know any alternatives, and why would We use those?
34. Explain what a single page app is and how to make one SEO-friendly.
35. What is the extent of We r experience with Promises and/or their polyfills?
36. What are the pros and cons of using Promises instead of callbacks?
37. What are some of the advantages/disadvantages of writing JavaScript code in a language that compiles to JavaScript?
38. What tools and techniques do We use debugging JavaScript code?
39. What language constructions do We use for iterating over object properties and array items?
40. Explain the difference between mutable and immutable objects.
    - What is an example of an immutable object in JavaScript?
    - What are the pros and cons of immutability?
    - How can We achieve immutability in We r own code?
41. Explain the difference between synchronous and asynchronous functions.
42. What is event loop?
    - What is the difference between call stack and task queue?
43. Explain the differences on the usage of `foo` between `function foo() {}` and `var foo = function() {}`
44. What are the differences between variables created using `let`, `var` or `const`?
45. What is javascript:void(0) ?

GENERAL

How do We optimize a website's assets?

- File concatenation, file compression, CDN Hosting, offloading assets, re-organizing and refining code, etc.

What are the ways to reduce page load time?

- Reduce image sizes, remove unnecessary widgets, HTTP compression, put CSS at the top and script references at the bottom or in external files, reduce lookups, minimize redirects, caching, etc.
- Remove Unnecessary fonts.
- avoid @import. (sequential load) (<link href> parallel load)
- avoid complex selector.
- Use sprite image
- Use minifier and gzip to decrease the page size.

What kind of things must We be aware of when design or developing for multilingual sites?

- setting the default language, using Unicode encoding, using the 'lang' attribute, being aware of standard font sizes and text direction, and language word length (may affect laWet).

Can We describeWe r workflow when We create a web page?

- Yet to Answer

Can We describe the difference between progressive enhancement and graceful degradation?

- Graceful degradation means building an application with a baseline of full functionality available in modern browsers and then taking the layers off to ensure it works with older browsers. Basically, We downgrade/degrade the enhanced version.

- Progressive enhancement is the opposite of graceful degradation. Instead of developing all features from the start, a web page is built from a baseline of the features supported by all browsers (and browser versions). Then, more advanced features are added like layers, so the web page takes advantage of the functionality newer browsers have to offer.

How many resources will a browser download from a given domain at a time?

- It depends on browser implementations. Usually 6 to 8 in the modern browsers, and less in the old browsers.

What does CORS stand for and what issue does it address?

- CORS stands for Cross Orign Resource Sharing and is used to get around the browsers same-origin policy. For security purposes, a browser won't load requests for resources to other domains when those requests are initated by scripts. CORS gets around this issue by supplying a special header that specifies which domains may make XMLHttpRequests for its resources.

How to handle cross-origin issues ?

- The browser sends the OPTIONS request with an Origin HTTP header. Ex: Origin: http://www.example.com
- The server at service.example.com may respond with:Access-Control-Allow-Origin: http://www.example.com or Access-Control-Allow-Origin: *

How to avoid cross-browser compatibility issues

- add <!DOCTYPE html> , it ensures a perfectly rendered site in every browser.
- HTML validator (W3C) And CSS validator (jigsaw)
- CSS Resets to remove the default design and apply custom design.
- Keep separate style sheets for every browser & use conditional statement in the HTML to invoke the right stylesheet for browser.
- add vendor specific functions (-moz,-ms,-webkit)
- use cross browser friendly frameworks (angular,react)

HTML

What does a **!DOCTYPE** do?

- The <!DOCTYPE> declaration is not an HTML tag; it is an instruction to the browser to inform about the version of html document and how browser should render it .

What happens when we don’t write DOCTYPE?

- new features & tags in HTML5 such as < article >,< footer >, < header >, <nav>, < section > may not be supported if the Doctype is not declared.
- It ensures how element should be displayed on the page by most of the browser. And it also makes browser's life easier. Otherwise, browser will guess and will go to quirks mode instead of standard mode.

DOCTYPEis case Sensitive?

- Yes. The DOCTYPE is case insensitive. The following DOCTYPEs are all valid:
  - <!doctype html>
  - <!DOCTYPE html>
  - <!DOCTYPE HTML>
  - <!DoCtYpEhTmL>

What is the use of data- attribute?

- allowWe to store extra information/ data in the DOM. u can write valid html with embedded private data. We can easily access data attribute by using javascript and hence a lot of libraries like knockout uses it.
- <div id="myDiv" data-user="jsDude" data-list-size="5" data-maxage="180"></div>

What all tags are important in head section of html file?

- <title> (this element is required in an HTML document)
- <style>
- <base>

- <link>
- <meta>
- <script>
- <noscript>

What is the use of <meta> tag?

- The <meta> tag provides metadata about the HTML document.
- The <meta> tag is used to describe the page in some way as well as other things such as refreshing a page automatically after a certain amount of time, and preventing webpages from being displayed in another websites frames page. Some of the things that can be described using the <meta> tag include the pages author, the software used to create the page, and a description of the content on the page.
- The <meta> tag does not have a closing tag.
- <meta charset="UTF-8">(Specify the character encoding for the HTML document)
- <meta name="description" content="Free Web tutorials on HTML and CSS">(Define a description of We r web page)
- <meta name="keywords" content="HTML, CSS, XML, XHTML, JavaScript">(Define keywords for search engines)
- <meta name="author" content="John Doe">(Define the author of a page)
- <meta http-equiv="refresh" content="30"> (Refresh document every 30 seconds)
- <meta name="viewport" content="width=device-width, initial-scale=1.0"> (Setting the viewport to make We r website look good on all devices)

What's the difference between full standards mode, almost standards mode and quirks mode?

- If we will write <!DOCTYPE html> the browser will be open in full standard mode.
- If we don’t write <!DOCTYPE html> the browser will be open in quick mode.

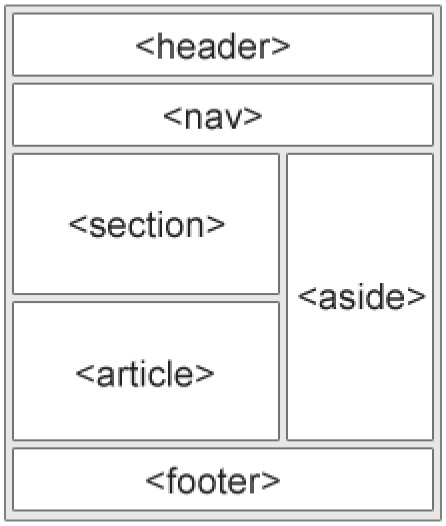
How do We serve a page with content in multiple languages?

- To set the primary language of our page as English we use the 'lang' attribute along with our 'en' language code and apply this to the HTML element at the beginning of each page.
- <html lang="en">

Describe the difference between <script>, <script async> and <script defer>

- <script>: The HTML file will be parsed until the script file is hit, at that point parsing will stop and a request will be made to fetch the file (if it's external). The script will then be executed before parsing is resumed.
- <scriptasync>: async downloads the file during HTML parsing and will pause the HTML parser to execute it when it has finished downloading.
- <script defer>: defer downloads the file during HTML parsing and will only execute it after the parser has completed. defer scripts are also guarenteed to execute in the order that they appear in the document.

What are the building blocks of HTML5?



Describe the difference between a cookie, sessionStorage and localStorage.

- sessionStorage, localStorage and Cookies all are used to store data on the client side. Each one has its own storage and expiration limit.

- localStorage: stores data with no expiration date, and gets cleared only through JavaScript, or clearing the Browser Cache / Locally Stored Data.

- sessionStorage: similar to localStorage but expires when the browser gets closed.

- Cookie: stores data that has to be sent back to the server with subsequent requests. Its expiration varies based on the type and the expiration duration can be set from either server-side or client-side (normally from server-side).

- Cookies are primarily for server-side reading (can also be read on client-side), localStorage and sessionStorage can only be read on client-side.

Why is it generally a good idea to position CSS <link>s between <head></head> and JS <script>s just before </body>? Do We know any exceptions?

- usually put the <link> tags in between the <head> to prevent Flash of Unstyled Content which gives the user something to look at while the rest of the page is being parsed
- Since Javascript blocks rendering by default, and the DOM and CSSOM construction can be also be delayed, it is usually best to keep scripts at the bottom of the page.

What is progressive rendering?

- Progressive rendering is a rendering mode in which the program gradually updates small parts of the entire image refining it from low quality to final result rather than focusing on one small part of the image at a time. The goal of progressive rendering is to always be able to see the process of refining the rendered image as a whole, pretty much like a painter sees his work evolve from a sketch by looking at the whole canvas after he adds more and more details.

Why We would use a srcset attribute in an image tag? Explain the process the browser uses when evaluating the content of this attribute.

- Srcset is a new attribute which allows We to specify different kind of images for different screen-sizes/orientation/display-types. The usage is really simple, We just provide a lot of different images separating them with a comma like this: <imgsrc="image.jpg" alt="image" srcset="<img><descriptor>, ..., <img_n><descriptor_n>">. Here is an example: srcset="image.jpg 160w, image2.jpg 320w, image3.jpg 2x"

Have We used different HTML templating languages before?

- Mustache
- Handlebars
- Dust
- EJS
- Underscore
- Pug
- ECT
- Template7
- JTemplates

Div vs Span

- div is a block element, span is inline.
- Width & height cann't be applied to span, where as width and height can be applied to div.

What is the difference between form get and form post?

- GET requests can be cached
- GET requests remain in the browser history
- GET requests can be bookmarked
- GET requests should never be used when dealing with sensitive data
- GET requests have length restrictions

- GET requests should be used only to retrieve data

- POST requests are never cached
- POST requests do not remain in the browser history
- POST requests cannot be bookmarked
- POST requests have no restrictions on data length

SVG vs Canvas

Canvas	**SVG**
• Resolution dependent • No support for event handlers • Poor text rendering capabilities • We can save the resulting image as .png or .jpg • Well suited for graphic-intensive games	• Resolution independent • Support for event handlers • Best suited for applications with large rendering areas (Google Maps) • Slow rendering if complex (anything that uses the DOM a lot will be slow) • Not suited for game applications

LocalStorage vs session storage?

- The localStorage object stores the data with no expiration date. The data will not be deleted when the browser is closed, and will be available the next day, week, or year.
- The sessionStorage object is equal to the localStorage object, **except** that it stores the data for only one session. The data is deleted when the user closes the specific browser tab.

What is the lifetime of local storage?

- There is no time limit for local storage, it doesn't have the expiration limit. User can set ut Manually

What is the difference between local storage and cookies?

- Cookies and local storage serve different purposes. Cookies are primarily for reading server-side, local storage can only be read by the client-side. So the question is, in We r app, who needs this data — the client or the server?

What are some of the key new features in HTML5?

- HTML 5 Semantic Elements
- New Form Types
- New form Elements
- Offline storage
- SVG & Canvas
- Audio & video
- REGEX

What are some of the key new APIs in HTML5?

- Drag & Drop
- Offline Storage
- Web-workers
- Application Cache

What is we-worker in HTML5?

- Web Worker
- A web worker is a JavaScript that runs in the background, independently of other scripts, without affecting the performance of the page. You can continue to do whatever you want: clicking, selecting things, etc., while the web worker runs in the background.

What is application cache in HTML5?

- Application cache
- **HTML5** introduces **application cache**, which means that a web **application** is **cached**, and accessible without an internet connection. **Application cache** gives an **application** three advantages: Offline browsing **-** users can use the **application** when they're offline. Speed **-** **cached** resources load faster.

CSS Questions:

What is the difference between classes and IDs in CSS?

- The difference between an ID and a class is that an ID can be used to identify one element, whereas a class can be used to identify more than one.
- Each element can have only one ID. Each page can have only one element with that ID
- We can use the same class on multiple elements.We can use multiple classes on the same element.

What's the difference between "resetting" and "normalizing" CSS? Which would We choose, and why?

- CSS resets aim to remove all built-in browser styling. Standard elements like H1-6, p, strong, em, etc end up looking exactly alike, having no decoration at all. For e.g. `margin`s, `padding`s, `font-size`s of all elements are reset to be the same. You're then supposed to add all decoration yourself.
- Normalizing preserves useful default styles rather than “unstyling” everything. Normalize CSS aims to make built-in browser styling consistent across browsers. Elements like H1-6 will appear bold, larger etc in a consistent way across browsers. You're then supposed to add only the difference in decoration your design needs. It also corrects bugs for common browser dependencies
- Normalize.css works for your target audience, then using Normalize.CSS instead of a CSS reset will make your own CSS smaller and faster to write.

Describe Floats and how they work.

46. Describe z-index and how stacking context is formed.
47. Describe BFC(Block Formatting Context) and how it works.
48. What are the various clearing techniques and which is appropriate for what context?
49. Explain CSS sprites, and how We would implement them on a page or site.
50. What are We r favourite image replacement techniques and which do We use when?
51. How would We approach fixing browser-specific styling issues?
52. How do We serveWe r pages for feature-constrained browsers?
    - What techniques/processes do We use?
53. What are the different ways to visually hide content (and make it available only for screen readers)?
54. Have We ever used a grid system, and if so, what do We prefer?
55. Have We used or implemented media queries or mobile specific laWets/CSS?
56. Are We familiar with styling SVG?
57. How do We optimizeWe r webpages for print?

58. What are some of the "gotchas" for writing efficient CSS?
59. What are the advantages/disadvantages of using CSS preprocessors?
    - Describe what We like and dislike about the CSS preprocessors We have used.
60. How would We implement a web design comp that uses non-standard fonts?
61. Explain how a browser determines what elements match a CSS selector.
62. Describe pseudo-elements and discuss what they are used for.
63. Explain We r understanding of the box model and how We would tell the browser in CSS to render We r laWet in different box models.
64. What does `* { box-sizing: border-box; }` do? What are its advantages?
65. List as many values for the display property that We can remember.
66. What's the difference between inline and inline-block?
67. What's the difference between a relative, fixed, absolute and statically positioned element?
68. The 'C' in CSS stands for Cascading. How is priority determined in assigning styles (a few examples)? How can We use this system to We r advantage?
69. What existing CSS frameworks have We used locally, or in production? How would We change/improve them?
70. Have We played around with the new CSS Flexbox or Grid specs?
71. How is responsive design different from adaptive design?
72. Have We ever worked with retina graphics? If so, when and what techniques did We use?
73. Is there any reason We 'd want to use `translate()` instead of *absolute positioning*, or vice-versa? And why?
74. What is the difference between inline, inline-block, and block?
75. What is the "Box Model" in CSS? Which CSS properties are a part of it?
76. Explain the difference between visibility:hidden; and display:none; ?
77. How do We clear a floated element?
78. Pseudo elements vs pseudo classes
79. How to align a div center both vertically and Horizontally.
80. What is a sprite? How is it applied using CSS? What is the benefit?
81. What are some accessibility concerns that come up in CSS?
82. What Is A CssReset. What Is The Difference Between A Css Reset And Normalize.css
83. Ans-CSS Reset removes browser default styles. Normalize.css sets a standard across all browsers (It does not 'reset' them)
84. How Would We Solve A Floated Div's Parent Height.
85. Ans-Clearfix, Float Parent as well, User overflow property other than 'visible'.
86. What tools do We use for cross-browser testing?
87. They should have some kind of strategy. Perhaps a web-based tool like BrowserStack. Perhaps a VM based tool like Virtual Box. Perhaps different actual computers. Part of the job of front end design is making sure things work everywhere they can (based on decided-upon support). We don't have to love it, but We can't hate it. "This right here, this is the job. What kind of work were We expecting?

JS Questions:

Explain event delegation

- Event delegation allows us to attach a single event listener, to a parent element, that will fire for all descendants matching a selector, whether those descendants exist now or are added in the future.
- There is no need to unbind the handler from elements that are removed and to bind the event for new elements.

Explain how this works in JavaScript

- When this is called outside of any function, in a global context, this defaults to the Window object in the browser.
- In the strict mode, value of 'this' will be undefined in the global scope.
- When we create a new instance of an object with the new keyword, this refers to the instance.
- this inside object's methodrefers to the object itself.
- this inside normal methodrefers to the window object.
- this is set to the element that fired the event in an event listener

Explain how prototypal inheritance works

46. Explain why the following doesn't work as an IIFE: `function foo(){ }();`.
    - o What needs to be changed to properly make it an IIFE?
47. What's the difference between a variable that is: `null`, `undefined` or undeclared?
    - o How would We go about checking for any of these states?
48. What is a closure, and how/why would We use one?
49. Can We describe the main difference between a `forEach` loop and a `.map()` loop and why We would pick one versus the other?
50. What's a typical use case for anonymous functions?
51. How do We organizeWe r code? (module pattern, classical inheritance?)
52. Explain the difference between classical inheritance and prototypal inheritance.
53. What's the difference between host objects and native objects?
54. Difference between: `function Person(){}`, `var person = Person()`, and `var person = new Person()`?
55. What's the difference between `.call` and `.apply`?
56. Explain `Function.prototype.bind`.
57. When would We use `document.write()`?
58. What's the difference between feature detection, feature inference, and using the UA string?
59. Explain Ajax in as much detail as possible.
60. What are the advantages and disadvantages of using Ajax?
61. Explain how JSONP works (and how it's not really Ajax).
62. Have We ever used JavaScript templating?
    - o If so, what libraries have We used?
63. Explain "hoisting".
64. Describe event bubbling.
65. What's the difference between an "attribute" and a "property"?
66. Why is extending built-in JavaScript objects not a good idea?

67. Difference between document load event and document DOMContentLoaded event?
68. What is the difference between == and ===?
69. Explain the same-origin policy with regards to JavaScript.
70. Make this work:
    - `duplicate([1,2,3,4,5]); // [1,2,3,4,5,1,2,3,4,5]`
71. Why is it called a Ternary expression, what does the word "Ternary" indicate?
72. What is `"use strict";`?what are the advantages and disadvantages to using it?
73. Create a for loop that iterates up to `100` while outputting **"fizz"** at multiples of `3`, **"buzz"** at multiples of `5` and **"fizzbuzz"** at multiples of `3` and `5`
74. Why is it, in general, a good idea to leave the global scope of a website as-is and never touch it?
75. Why would We use something like the `load` event? Does this event have disadvantages? Do We know any alternatives, and why would We use those?
76. Explain what a single page app is and how to make one SEO-friendly.
77. What is the extent of We r experience with Promises and/or their polyfills?
78. What are the pros and cons of using Promises instead of callbacks?
79. What are some of the advantages/disadvantages of writing JavaScript code in a language that compiles to JavaScript?
80. What tools and techniques do We use debugging JavaScript code?
81. What language constructions do We use for iterating over object properties and array items?
82. Explain the difference between mutable and immutable objects.
    - What is an example of an immutable object in JavaScript?
    - What are the pros and cons of immutability?
    - How can We achieve immutability in We r own code?
83. Explain the difference between synchronous and asynchronous functions.
84. What is event loop?
    - What is the difference between call stack and task queue?
85. Explain the differences on the usage of `foo` between `function foo() {}` and `var foo = function() {}`
86. What are the differences between variables created using `let`, `var` or `const`?
87. What is javascript:void(0) ?