marko-jankovic / **CSS and HTML interview questions.md**

Last active 10 days ago • Report abuse

Embed ▾ `<script src="https://gist.` Download ZIP

CSS and HTML interview questions

`CSS and HTML interview questions.md`

CSS

What is CSS?

- CSS stands for Cascading Style Sheet.
- Styles define how to display HTML elements
- Styles were added to HTML 4.0 to solve a problem
- External Style Sheets can save a lot of work
- External Style Sheets are stored in CSS files

Where to define styles? How can you integrate/import CSS on a web page?

- Inline, used to style only a small piece of code `<p style="color:blue"> Hello CSS </p>`
- Internal/Embedded, style sheets are put between the ... `<style> p{color:blue} </style>`
- External `<link rel="stylesheet" type="text/css" href="style.css"`

What is Property?

- The style that you are applying to a selector, e.g. border.

What is Selector?

The way you declare which elements the styles should apply to. There are different kinds of selectors:

- `Class` : The most commonly used selector. E.g. ".cloudy" to select an element with classname cloudy. There can be more than 1 element with the same classname.
- `ID` : Use this sparingly. You cannot reuse an ID within the same page and used only to identify an element uniquely. E.g. `<div id=lovelyweather></div>`
- `Attribute Selector` : If you use any attribute other than class or id to identify an element in a stylesheet, you would be using Attribute Selectors. You can also do basic pattern matching within an attribute selector (so if you would like to do basic pattern matching for selectors using class or ID attributes, you would want to use attribute selectors).
- `Pseudo-Classes` : Classes that are applied to elements based on information that is not present in the markup, e.g. :first-child or :last-child. Do note that the selectors are parsed from right to left (see the demo). You cannot use section `article:first-child` to select the first occurrence of article, if the first child of section is h1 and not article. Likewise with the :nth-child, and :last-child pseudo-classes.
- `Pseudo-Elements` : Pseudo-elements differ from Pseudo-Classes in that they actually create an element in the document tree. This is almost the first instance of CSS modifying the HTML document tree. You should ideally use pseudo-elements with "::" instead of ":" (but most browsers accept ":" notation for CSS 2.1 pseudo-elements). Pseudo-elements are: `::first-line` , `::first-letter` , `::before` , `::after` (See the demo for how pseudo-elements work).

What are Combinators?

The selection of an element based on its occurrence in relation to another element (chosen by the choice of combinator: whitespace, >, +, or ~). You can have:

- Descendant Combinator
  - This is the most common usage, e.g. #lovelyweather h1.
- Child Combinator
  - Select an element if it is a direct child of another element (and not a grandchild of that element).
- Adjacent Sibling Combinator
  - The element that is immediately adjacent to another element.
- General Sibling Combinator
  - The element that is adjacent, but not immediately to another element.

Why background and color are the separate properties if they should always be set together?

Color is an inherited property while background is not. So this can make confusion further.

What is the difference between class selectors and id selectors?

An overall block is given to class selector while id selectors take only a single element differing from other element

When working on a large codebase CSS it can quickly become unwieldly and difficult to maintain. How do you approach this problem? How do you architect your CSS (and have you heard/used BEM, OOCS or SMACSS)?

What is CSS BEM

The BEM methodology is a naming convention for CSS classes in order to keep CSS more maintainable by defining namespaces to solve scoping issues. BEM stands for Block Element Modifier which is an explanation for its structure. A Block is a standalone component that is reusable across projects and acts as a "namespace" for sub components (Elements). Modifiers are used as flags when a Block or Element is in a certain state or is different in structure or style.

```
/* block component */
.block {
}

/* element */
.block__element {
}

/* modifier */
.block__element--modifier {
}
```

Here is an example with the class names on markup:

```
<nav class="navbar">
  <a href="/" class="navbar__link navbar__link--active"></a>
  <a href="/" class="navbar__link"></a>
  <a href="/" class="navbar__link"></a>
</nav>
```

In this case, `navbar` is the Block, `navbar__link` is an Element that makes no sense outside of the navbar component, and `navbar__link--active` is a Modifier that indicates a different state for the `navbar__link` Element.

Since Modifiers are verbose, many opt to use `is-*` flags instead as modifiers.

```
<a href="/" class="navbar__link is-active"></a>
```

These must be chained to the Element and never alone however, or there will be scope issues.

```
.navbar__link.is-active {
}
```

References

```
* https://css-tricks.com/bem-101/
```

How do you organize CSS files?

What are the advantages of External Style Sheets?

- You can create classes for reusing it in many documents.
- By using it, you can control the styles of multiple documents from one file.
- In complex situations, you can use selectors and grouping methods to apply styles.

What is RWD (Responsive web design) ?

This technique is used to display the designed page perfectly on every screen size and device. For example: Mobile, Tablet, desktop, laptop etc. You don't need to create a different page for each device.

Explain CSS sprites, and how you would implement them on a page or site. How do you go about creating them? What are possible alternatives to sprites?

CSS sprites combine multiple images into one single larger image. It is a commonly-used technique for icons. How to implement it:

- Use a sprite generator that packs multiple images into one and generate the appropriate CSS for it.
- Each image would have a corresponding CSS class with `background-image` , `background-position` and `background-size` properties defined.
- To use that image, add the corresponding class to your element.

Advantages:

- Reduce the number of `HTTP` requests for multiple images (only one single request is required per spritesheet). But with `HTTP2` , loading multiple images is no longer much of an issue.
- Advance downloading of assets that won't be downloaded until needed, such as images that only appear upon `:hover` pseudo-states. Blinking wouldn't be seen.
- When you have multiple images/ icons, browser makes separate call to the server for each one of them. sprite is a technique to combine all/ some of them (usually similar one in terms of type of image. For example, you will put jpg in one sprite) in one image. To display the icon you set height, width and background position.

Alternatives:

- Data URIs - allow you to embed the image data directly into a stylesheet. This avoids additional HTTP requests for images, making it essentially the same thing as a sprite, without the fancy positioning.
- Icon Fonts
- SVGs

References

- https://css-tricks.com/css-sprites/
- https://css-tricks.com/spriting-img/
- http://css-tricks.com/data-uris/

What are the benefits of CSS sprites?

If a web page has large no. of images that takes a longer time to load because each image separately sends out an `HTTP` request. The concept of CSS sprites is used to reduce the loading time for a web page because it combines the various small images into one image. It reduces the number of `HTTP` requests and hence the loading time.

CSS sprites combine multiple images into one image, limiting the number of HTTP requests a browser has to make, thus improving load times. Even under the new HTTP/2 protocol, this remains true.

Under HTTP/1.1, at most one request is allowed per TCP connection. With HTTP/1.1, modern browsers open multiple parallel connections (between 2 to 8) but it is limited. With HTTP/2, all requests between the browser and the server are multiplexed on a single TCP connection. This means the cost of opening and closing multiple connections is mitigated, resulting in a better usage of the TCP connection and limits the impact of latency between the client and server. It could then become possible to load tens of images in parallel on the same TCP connection.

However, according to benchmark results, although HTTP/2 offers 50% improvement over HTTP/1.1, in most cases the sprite set is still faster to load than individual images.

To utilize a spritesheet in CSS, one would use certain properties, such as `background-image`, `background-position` and `background-size` to ultimately alter the background of an element.

What is the CSS Box model and what are its elements?

- The CSS box model is used to define the design and layout of elements of CSS.
- The elements are:
  - Margin
  - Border
  - Padding
  - Content

What is the float property and what float do.

- The CSS float property is used to move the image to the right or left along with the texts to be wrapped around it. It doesn't change the property of the elements used before it
- float pushes an element to the sides of a page with text wrapped around it. you can create entire page or a smaller area by using float. if size of a floated element changes, text around it will re-flow to accommodate the changes. You can have float left, right, none or inherit.
- if you set, 'float: left;' for an image, it will move to the left until the margin, padding or border of another block-level element is reached. The normal flow will wrap around on the right side.

How can you clear sides of a floating element?

If you clear a slide of an element, floating elements will not be accepted on that side. With 'clear' set to 'left', an element will be moved below any floating element on the left side. clear is used to stop wrap of an element around a floating element.

What is tweening?

- It is the process of generating intermediate frames between two images.
- It gives the impression that the first image has smoothly evolved into the second one.
- It is an important method used in all types of animations.
- In CSS3, Transforms (matrix, translate, rotate, scale etc.) module can be used to achieve tweening.

Who maintains the CSS specifications? What do you understand by W3C?

W3C stands for World Wide Web Consortium. The mission of the W3C is to lead the Web to its full potential by developing relevant protocols and guidelines.This is achieved primarily by creating and publishing Web standards. By adopting the Web standards created by the W3C, hardware manufacturers and software developers can ensure their equipment and programs work with the latest Web technologies. For example, most Web browsers incorporate several W3C standards, which allows them to interpret the latest versions of HTML and CSS code. When browsers conform to the W3C standards, it also helps Web pages appear consistent across different browsers.

Explain the difference between `visibility: hidden;` and `display: none;` ? What are the pros and cons of using `display:none` ?

- `visibility: hidden` simply hides the element but it will occupy space and affect the layout of the document.
- `display: none` removes the element from the normal layout flow (causes DOM reflow). It will not affect the layout of the document nor occupy space.

What is re-flow?

How does `z-index` relate to positioning? Describe `z-index` and how stacking context is formed?

- The `z-index` property specifies the stack order of elements. An element with a higher `z-index` stack order is always rendered in front of an element with a lower `z-index` stack order on the screen. `z-index` only works on positioned elements `position: absolute` , `position: relative` , or `position: fixed` . The default stack order of non-positioned elements is their order in the document.

What is the purpose of the `z-index` and how is it used?

- The `z-index` helps specify the stack order of positioned elements that may overlap one another. The `z-index` default value is zero, and can take on either a positive or negative number.
- An element with a higher `z-index` is always stacked above than a lower index.
- `z-index` can take the following values:
  - Auto: Sets the stack order equal to its parents.
  - Number: Orders the stack order.
  - Initial: Sets this property to its default value (0).
  - Inherit: Inherits this property from its parent element.

What existing CSS frameworks have you used locally, or in production? How would you change/improve them?

What are CSS frameworks? What CSS frameworks have you used

It is a pre-planned libraries, which allows easier and more standards-compliant webpage styling, using CSS language.

How works absolute / relative / fixed / static position?

`absolute` , place an element exactly where you want to place it. absolute position is actually set relative to the element's parent. if no parent available then relatively place to the page itself (it will default all the way back up to the element).

`relative` , means "relative to itself". Setting position: relative; on an element and no other positioning attributes, it will no effect on it's positioning. It allows the use of `z-index` on the element and it limits the scope of absolutely positioned child elements. Any child element will be absolutely positioned within that block.

`fixed` , element is positioned relative to viewport or the browser window itself. viewport doesn't changed if u scroll and hence fixed element will stay right in the same position.

- `static` default for every single page element. The only reason you would ever set an element to position: static is to forcefully-remove some positioning that got applied to an element outside of your control.

- `sticky` - Sticky positioning is a hybrid of relative and fixed positioning. The element is treated as `relative` positioned until it crosses a specified threshold, at which point it is treated as `fixed` positioned.

References

```
* https://css-tricks.com/absolute-relative-fixed-positioining-how-do-they-differ/
```

How is behave absolute element if it is inside fixed element/relative/absolute element

What are the pros and cons of using absolute positioning?

The difference between block / inline / inline-block element

- Elements with `display: inline-block` are like `display: inline elements`, but they can have a width and a height. That means that you can use an `inline-block` element as a block while flowing it within text or other elements.
- `block`:
  - respect all of those
  - force a line break after the block element
  - breaks the flow
- `inline`:
  - respect left & right margins and padding, but not top & bottom
  - cannot have a width and height set
  - margin and padding will push other elements horizontally not vertically
  - allow other elements to sit to their left and right.
  - elements do not break the flow
- `inline-block`:
  - allow other elements to sit to their left and right
  - respect top & bottom margins and padding
  - respect height and width

List CSS selectors and their priorities

- https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Selectors

Talk about position property, and its values

What are counters in CSS3 ?

How to hide an element?

Are CSS rule names case sensitive?

- No

Multiple Class / ID and Class Selectors

- https://css-tricks.com/multiple-class-id-selectors

Why CSS selectors mixed up with cases (uppercase and lowercase) don't apply the styles?

- because HTML IDs and Classes are case sensetive.

Does margin-top or margin-bottom has effect on inline element?

- No

Does padding-top or padding-bottom has effect on inline element?

- No

Which one would you prefer among px, em % or pt and why?

it depends on what you are trying to do.

- `px` gives fine grained control and maintains alignment because 1 px or multiple of 1 px is guaranteed to look sharp. px is not cascade, this means if parent font-size is 20px and child 16px. child would be 16px.
- `em` maintains relative size. you can have responsive fonts. em is the width of the letter 'm' in the selected typeface. However, this concept is tricky. 1em is equal to the current font-size of the element or the browser default. if u sent font-size to 16px then 1em = 16px. The common practice is to set default body font-size to 62.5% (equal to 10px). em is cascade
- `%` sets font-size relative to the font size of the body. Hence, you have to set font-size of the body to a reasonable size. this is easy to use and does cascade. for example, if parent font-size is 20px and child font-size is 50%. child would be 10px.
- `pt` (points) are traditionally used in print. 1pt = 1/72 inch and it is fixed-size unit.

Does padding-left or padding-right or margin-left or margin-right has effect on inline element?

- Yes

**If you have a
element with font-size: 10rem, will the text be responsive when the user resizes / drags the browser window?**

- No

Describe pseudo-elements and discuss what they are used for.

- https://developer.mozilla.org/en-US/docs/Web/CSS/pseudo-elements
- A CSS pseudo-element is used to style specified parts of an element.

For example, it can be used to:

```
Style the first letter, or line, of an element
Insert content before, or after, the content of an element
```

```
p::first-line {
    color: #ff0000;
    font-variant: small-caps;
}
```

In a HTML document, the pseudo class :root always refers to the element

- True

What is pseudo element? what is pseudo class?

- pseudo elements helps you to add cosmetics contents. pseudo elements generates content where as pseudo class deals with state of the element. for example, you can style `:first-letter` of every paragraph. similarly, `:first-line` and fancy stuff with `:before` , `:after`

The pseudo class :checked will select inputs with type radio or checkbox, but not elements.

- False

How to center align a paragraph?

How to center align a div inside another div?

How to make a two column Web page? a three column Web page?

How to draw a triangle? a circle? a colored square?

How to make a tab view?

How do you define multiple background images through CSS?

Explain the CSS “box model” and the layout components that it consists of. Provide some usage examples.

- The CSS box model is a rectangular layout paradigm for HTML elements that consists of the following:

  `Content` The content of the box, where text and images appear

  `Padding` A transparent area surrounding the content (i.e., the amount of space between the border and the content)

  `Border` A border surrounding the padding (if any) and content

  `Margin` A transparent area surrounding the border (i.e., the amount of space between the border and any neighboring elements)

```
/* top   right  bottom left  */
padding: 25px  50px   75px   100px;

/* same padding on all 4 sides: */
padding: 25px;

/* top/bottom padding 25px; right/left padding 50px */
padding: 25px 50px;

/* top padding 25px; right/left padding 50px; bottom padding 75px */
padding: 25px 50px 75px;
```

Explain what elements will match each of the following CSS selectors:

- div, `p` Selects all `<div>` elements and all

  elements

- div `p` Selects all `<p>` elements that are anywhere inside a element

- div > `p` Selects all `<p>` elements where the immediate parent is a element
- div + `p` Selects all `<p>` elements that are placed immediately after a element
- div ~ `p` Selects all `<p>` elements that are anywhere preceded by a element

Explain the meaning of each of these CSS units for expressing length:

`cm` centimeters `em` elements (i.e., relative to the font-size of the element; e.g., 2 em means 2 times the current font size) `in` inches `mm` millimeters `pc` picas (1 pc = 12 pt = 1/6th of an inch) `pt` points (1 pt = 1/72nd of an inch) `px` pixels (1 px = 1/96th of an inch)

Which values (units) could be used for width?

In CSS3, how would you select:

- Every `<a>` element whose href attribute value begins with “https”.
  - a[href^="https"]
- Every `<a>` element whose href attribute value ends with “.pdf”.
  - a[href$=".pdf"]
- Every `<a>` element whose href attribute value contains the substring “css”.
  - a[href*="css"]

How does CSS actually work (under the hood of browser)?

```
https://github.com/FAQGURU/FAQGURU/blob/master/topics/en/css.md#how-does-css-actually-work-under-the-hood-of-browser
```

What is the purpose of the box-sizing property? What are it's possible values?

Provide an example of content-box vs border-box box-sizing

How to create a zebra striped table with CSS?

To create a zebra-striped table, use the nth-child() selector and add a background-color to all even (or odd) table rows:

```
tr:nth-child(even) {
    background-color: #f2f2f2
}
```

What is the difference between RGBa and HSLa? When would you use one vs the other?

What libraries and tools you are using?

What preprocessors you are using?

What are the reasons to use preprocessor

- you write css in high level with some special syntax (declaring variable, nested syntax, mathematical operations, etc.) and that is compiled to css. Preprocessor helps you to speed up develop, maintain, ensure best practices and also confirms concatenation, compression, etc.

What is CSS preprocessor and why to user one?

- A CSS preprocessor is a program that lets you generate CSS from the preprocessor's own unique syntax. There are many CSS preprocessors to choose from, however most CSS preprocessors will add some features that don't exist in pure CSS, such as mixin, nesting selector, inheritance selector, and so on. These features make the CSS structure more readable and easier to maintain.
- Here are a few of the most popular CSS preprocessors:
  - SASS (SCSS)
  - LESS
  - Stylus
  - PostCSS

What are the advantages/disadvantages of using CSS preprocessors?

Advantages:

```
* CSS is made more maintainable.
* Easy to write nested selectors.
* Variables for consistent theming. Can share theme files across different projects.
* Mixins to generate repeated CSS.
* Splitting your code into multiple files. CSS files can be split up too but doing so will require an HTTP
request to download each CSS file.
```

Disadvantages:

```
* Requires tools for preprocessing. Re-compilation time can be slow.
```

- Advantages of SASS/LESS: Use of variables, mixins, nesting, looping, partials and more.
- Advantages POSTCSS: Use of future syntax. Ability to select only the plugins you need for a particular project. Writing plain CSS and applying the plugin to output necessary syntax that suits your needs – which in turn allows for the ability to remove or update that plugin if at any time it is not useful anymore. In other words, POSTCSS offers a lot of flexibility AND all of the features SASS/LESS offer. Versions of POSTCSS have been developed for both Grunt and Gulp.
- Disadvantage POSTCSS: Keeping track of plugins for older projects could get hairy, though you have a package.json to reference.
- Disdvantages of SASS/LESS: They are monolithic, and you need to learn a new language.

What preprocessor do you use? (Sass, LESS, Stylus) Why do people use them? How does something like Compass relate to Sass?

- They are CSS preprocessors. They are a special syntax/language that compile down into CSS. They make managing CSS easier, with things like variables and mixins to handle vendor prefixes (among other things). They make doing best practices easier, like concatenating and compressing CSS.

Describe what you like and dislike about the CSS preprocessors you have used.

Likes:

- Mostly the advantages mentioned above.

- Less is written in JavaScript, which plays well with Node.

Dislikes:

- I use Sass via `node-sass`, which is a binding for LibSass written in C++. I have to frequently recompile it when switching between node versions.
- In Less, variable names are prefixed with `@`, which can be confused with native CSS keywords like `@media`, `@import` and `@font-face` rule.

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

- Resetting - Resetting is meant to strip all default browser styling on elements. For e.g. margins, paddings, font-sizes of all elements are reset to be the same. You will have to redeclare styling for common typographic elements.

- Normalizing - Normalizing preserves useful default styles rather than "unstyling" everything. It also corrects bugs for common browser dependencies.

Describe Floats and how they work.

Float is a CSS positioning property. Floated elements remain a part of the flow of the page, and will affect the positioning of other elements (e.g. text will flow around floated elements), unlike `position: absolute` elements, which are removed from the flow of the page.

The CSS `clear` property can be used to be positioned below `left` / `right` / `both` floated elements.

If a parent element contains nothing but floated elements, its height will be collapsed to nothing. It can be fixed by clearing the float after the floated elements in the container but before the close of the container.

The `.clearfix` hack uses a clever CSS pseudo selector (`:after`) to clear floats. Rather than setting the overflow on the parent, you apply an additional class `clearfix` to it. Then apply this CSS:

```css
.clearfix:after {
  content: ' ';
  visibility: hidden;
  display: block;
  height: 0;
  clear: both;
}
```

Alternatively, give `overflow: auto` or `overflow: hidden` property to the parent element which will establish a new block formatting context inside the children and it will expand to contain its children.

References

- https://css-tricks.com/all-about-floats/

Describe BFC(Block Formatting Context) and how it works.

A Block Formatting Context (BFC) is part of the visual CSS rendering of a web page in which block boxes are laid out. Floats, absolutely positioned elements, `inline-blocks`, `table-cells`, `table-caption` s, and elements with `overflow` other than `visible` (except when that value has been propagated to the viewport) establish new block formatting contexts.

A BFC is an HTML box that satisfies at least one of the following conditions:

- The value of `float` is not `none`.
- The value of `position` is neither `static` nor `relative`.
- The value of `display` is `table-cell`, `table-caption`, `inline-block`, `flex`, or `inline-flex`.
- The value of `overflow` is not `visible`.

In a BFC, each box's left outer edge touches the left edge of the containing block (for right-to-left formatting, right edges touch).

Vertical margins between adjacent block-level boxes in a BFC collapse. Read more on collapsing margins.

References

- https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Block_formatting_context
- https://www.sitepoint.com/understanding-block-formatting-contexts-in-css/

What are the various clearing techniques and which is appropriate for what context? How works .clearfix?

- Empty `div` method - `<div style="clear:both;"></div>`.
- Clearfix method - Refer to the `.clearfix` class above.
- `overflow: auto` or `overflow: hidden` method - Parent will establish a new block formatting context and expand to contains its floated children.

In large projects, I would write a utility `.clearfix` class and use them in places where I need it. `overflow: hidden` might clip children if the children is taller than the parent and is not very ideal.

What are your favourite image replacement techniques and which do you use when?

Does `overflow: hidden` create a new block formatting context?

- Yes. overflow property deals with the content if content size exceeds the allocated size for the content. You can make extra content visible, hidden, scroll or auto (viewport default behavior).

How would you approach fixing browser-specific styling issues?

- After identifying the issue and the offending browser, use a separate style sheet that only loads when that specific browser is being used. This technique requires server-side rendering though.
- Use `autoprefixer` to automatically add vendor prefixes to your code.
- Use Reset CSS or Normalize.css.

What are your favorite image replacement techniques and which do you use when?

- lazy loading.

- progressive image loading placeholders.

What techniques/processes do you use for building sites?

How do you serve your pages for feature-constrained browsers? What techniques/processes do you use?

- Graceful degradation - The practice of building an application for modern browsers while ensuring it remains functional in older browsers.
- Progressive enhancement - The practice of building an application for a base level of user experience, but adding functional enhancements when a browser supports it.
- Use caniuse.com to check for feature support.
- Autoprefixer for automatic vendor prefix insertion.
- Feature detection using Modernizr.
- Use CSS Feature queries @support

What are the different ways to visually hide content (and make it available only for screen readers)?

These techniques are related to accessibility (a11y).

- `visibility: hidden`. However, the element is still in the flow of the page, and still takes up space.
- `width: 0; height: 0`. Make the element not take up any space on the screen at all, resulting in not showing it.
- `position: absolute; left: -99999px`. Position it outside of the screen.

- `text-indent: -9999px` . This only works on text within the `block` elements.
- Metadata. For example by using Schema.org, RDF, and JSON-LD.
- WAI-ARIA. A W3C technical specification that specifies how to increase the accessibility of web pages.

Even if WAI-ARIA is the ideal solution, I would go with the `absolute` positioning approach, as it has the least caveats, works for most elements and it's an easy technique.

References

- https://www.w3.org/TR/wai-aria-1.1/
- https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA
- http://a11yproject.com/

Have you ever used a grid system, and if so, what do you prefer? Explain how a grid system works

I like the `float` -based grid system because it still has the most browser support among the alternative existing systems (flex, grid). It has been used in Bootstrap for years and has been proven to work.

Have you used or implemented media queries or mobile specific layouts/CSS?

How could you apply css rules specific to a media?

- `@media (max-width: 700px){...}` means you want to apply rules to those media whose max-width is 700 px. this means every smaller device will have this rule.

`@media (max-width: 700px) and (orientation: landscape){...}` will apply rules for media smaller than 700px and in landscape orientation.

What is the use of `@media only screen` ?

- hide style sheets from older user agents.

https://stackoverflow.com/questions/8549529/what-is-the-difference-between-screen-and-only-screen-in-media-queries

Can you name the four types of `@media` properties?

- `all` , which applies to all media type devices
- `print` , which only applies to printers
- `screen` , which only applies to screens (desktops, tablets, mobile etc.)
- `speech` , which only applies to screenreaders

Does the `screen` keyword apply to the device's physical screen or the browser's viewport?

- Browser's Viewport

Are you familiar with styling SVG?

Yes, there are several ways to color shapes (including specifying attributes on the object) using inline CSS, an embedded CSS section, or an external CSS file. Most SVG you'll find around the web use inline CSS, but there are advantages and disadvantages associated with each type.

Basic coloring can be done by setting two attributes on the node: `fill` and `stroke` . `fill` sets the color inside the object and `stroke` sets the color of the line drawn around the object. You can use the same CSS color naming schemes that you use in HTML, whether that's color names (that is `red`), RGB values (that is `rgb(255,0,0)`), Hex values, RGBA values, etc.

```
<rect x="10" y="10" width="100" height="100" stroke="blue"
  fill="purple" fill-opacity="0.5" stroke-opacity="0.8"/>
```

References

- https://developer.mozilla.org/en-US/docs/Web/SVG/Tutorial/Fills_and_Strokes

Can you give an example of an @media property other than screen?

Yes, there are four types of @media properties (including *screen*):

- `all` - for all media type devices
- `print` - for printers
- `speech` - for screenreaders that "reads" the page out loud
- `screen` - for computer screens, tablets, smart-phones etc.

Here is an example of `print` media type's usage:

```
@media print {
  body {
    color: black;
  }
}
```

References

- https://developer.mozilla.org/en-US/docs/Web/CSS/@media#Syntax

How do you optimize your webpages for print?

Remove and reduce colors to grayscale and black and whites if possible. Also printing gray's can reduce the amount of black ink needed – so that's a bonus if you want to really be nice. Add styling if you want the URLs to print out:

How would you implement a web design comp that uses non-standard fonts?

Use `@font-face` and define `font-family` for different `font-weight` s.

Explain how a browser determines what elements match a CSS selector.

This part is related to the above about writing efficient CSS. Browsers match selectors from rightmost (key selector) to left. Browsers filter out elements in the DOM according to the key selector and traverse up its parent elements to determine matches. The shorter the length of the selector chain, the faster the browser can determine if that element matches the selector.

For example with this selector `p span` , browsers firstly find all the `<span>` elements and traverse up its parent all the way up to the root to find the `<p>` element. For a particular `<span>` , as soon as it finds a `<p>` , it knows that the `<span>` matches and can stop its matching.

What are the some pseudo classed u have used?

- pseudo class tells you specific state of an element. allow to style element dynamically. The most popular one is `:hover` . Besides i have used :visited, `:focus` , `:nth-child` , `nth-of-type` , `:link` , etc.

- pseudo classes is better if you don't want to mess up with javaScript however, pseudo-classes is slow to process and apply rules.

How do you optimize css selectors?

- This is very open and depend on what you are trying to achieve. If i order selectors in terms of render speed it would be like id, class, tag, siblings, child, descendant, universal, attribute, pseudo. Speed of ID and class is very close. However your code should be readable, maintainable and DRY along with highly performant.

- The best practices in general are: avoid universal selectors, don't repeat yourself, remove redundant selectors, be as specific as possible, and keep learning.

How can you load css resources conditionally?

`@import` allows you to load/import stylesheet by using a path (uri) representing the location of the file. You can define one or more media by comma separation for which you want to load the stylesheet. If browser dont support the media stylesheet will not be loaded.

Describe pseudo-elements and discuss what they are used for.

A CSS pseudo-element is a keyword added to a selector that lets you style a specific part of the selected element(s). They can be used for decoration (`:first-line` , `:first-letter`) or adding elements to the markup (combined with `content: ...`) without having to modify the markup (`:before` , `:after`).

- `:first-line` and `:first-letter` can be used to decorate text.
- Used in the `.clearfix` hack as shown above to add a zero-space element with `clear: both` .
- Triangular arrows in tooltips use `:before` and `:after` . Encourages separation of concerns because the triangle is considered part of styling and not really the DOM. It's not really possible to draw a triangle with just CSS styles without using an additional HTML element.

References

- https://css-tricks.com/almanac/selectors/a/after-and-before/

What are the properties related to box model

- Technically, height, width, padding and border are part of box model and margin is related to it.

- Everything in a web page is a box where you can control size, position, background, etc. Each box/ content area is optionally surrounded by padding, border and margin. When you set height and width of an element, you set content height and width.

Explain your understanding of the box model and how you would tell the browser in CSS to render your layout in different box models.

- The content of the space taken by an element. Each element has an inner and outer height/width calculated based on the content, padding, border and margin.

  `content-box` default. Excludes padding, border and margin from the inner dimensions. `border-box` includes the padding and border, but not the margin in the inner dimension.

The CSS box model describes the rectangular boxes that are generated for elements in the document tree and laid out according to the visual formatting model. Each box has a content area (e.g. text, an image, etc.) and optional surrounding `padding` , `border` , and `margin` areas.

The CSS box model is responsible for calculating:

- How much space a block element takes up.
- Whether or not borders and/or margins overlap, or collapse.
- A box's dimensions.

The box model has the following rules:

- The dimensions of a block element are calculated by `width` , `height` , `padding` , `border` s, and `margin` s.
- If no `height` is specified, a block element will be as high as the content it contains, plus `padding` (unless there are floats, for which see below).
- If no `width` is specified, a non-floated block element will expand to fit the width of its parent minus `padding` .
- The `height` of an element is calculated by the content's `height` .
- The `width` of an element is calculated by the content's `width` .

- By default, `padding` s and `border` s are not part of the `width` and `height` of an element.

References

- https://www.smashingmagazine.com/2010/06/the-principles-of-cross-browser-css-coding/#understand-the-css-box-model

What does `* { box-sizing: border-box; }` do? What are its advantages?

- Make every element in the document include the padding and border in the element's inner dimensions; making it easier to reason about the layout of elements on the page.
- By default, elements have `box-sizing: content-box` applied, and only the content size is being accounted for.
- `box-sizing: border-box` changes how the `width` and `height` of elements are being calculated, `border` and `padding` are also being included in the calculation.
- The `height` of an element is now calculated by the content's `height` + vertical `padding` + vertical `border` width.
- The `width` of an element is now calculated by the content's `width` + horizontal `padding` + horizontal `border` width.
- Taking into account `padding` s and `border` s as part of our box model resonates better with how designers actually imagine content in grids.

References

- https://www.paulirish.com/2012/box-sizing-border-box-ftw/

List as many values for the display property that you can remember.

- `none` , `block` , `inline` , `inline-block` , `table` , `table-row` , `table-cell` , `list-item` .

The 'C' in CSS stands for Cascading. How is priority determined in assigning styles (a few examples)? How can you use this system to your advantage?

Can you explain the differences between em, rem, px?

What's `#target` .

What existing CSS frameworks have you used locally, or in production? How would you change/improve them?

What are most common tested browsers and devices?

Have you played around with the new CSS Flexbox or Grid specs?

- Flexbox or Flexible boxes, is a new layout mode in CSS3 Use of flexbox ensures that elements behave predictably when the page layout must accommodate different screen sizes and different display devices.
- For many applications, the flexible box model provides an improvement over the block model in that it does not use floats, nor do the flex container's margins collapse with the margins of its contents.
- Grid specs CSS Grid Layout is a specification for creating two-dimensional grids Grid is a companion to the Flexible Box Module (flexbox). Flexbox is designed for one-dimensional layout, so things can be arranged in an unbroken line. Grid is designed for two-dimensional layout, meaning the items don't need to sit next to each other. In the future we're likely to use both: Grid Layout for main page areas, and flexbox for the smaller UI elements it excels with.

How is responsive design different from adaptive design? What is responsive design? What is the difference between fixed and fluid layouts? What are some of the pros and cons with these designs?

Both responsive and adaptive design attempt to optimize the user experience across different devices, adjusting for different viewport sizes, resolutions, usage contexts, control mechanisms, and so on.

Responsive design works on the principle of flexibility - a single fluid website that can look good on any device. Responsive websites use media queries, flexible grids, and responsive images to create a user experience that flexes and changes based on a multitude of factors. Like a single ball growing or shrinking to fit through several different hoops.

Adaptive design is more like the modern definition of progressive enhancement. Instead of one flexible design, adaptive design detects the device and other features and then provides the appropriate feature and layout based on a predefined set of viewport sizes and other characteristics. The site detects the type of device used and delivers the pre-set layout for that device. Instead of a single ball going through several different-sized hoops, you'd have several different balls to use depending on the hoop size.

References

- https://developer.mozilla.org/en-US/docs/Archive/Apps/Design/UI_layout_basics/Responsive_design_versus_adaptive_design
- http://mediumwell.com/responsive-adaptive-mobile/
- https://css-tricks.com/the-difference-between-responsive-and-adaptive-design/

What is aspect ratio?

```
https://css-tricks.com/aspect-ratio-boxes

https://justmarkup.com/log/2015/11/definining-aspect-ratio-to-prevent-reflow
```

Have you ever worked with retina graphics? If so, when and what techniques did you use? What kind of techniques do you use to handle images for retina screens?

Retina is just a marketing term to refer to high resolution screens with a pixel ratio bigger than 1. The key thing to know is that using a pixel ratio means these displays are emulating a lower resolution screen in order to show elements with the same size. Nowadays we consider all mobile devices *retina* defacto displays.

Browsers by default render DOM elements according to the device resolution, except for images.

In order to have crisp, good-looking graphics that make the best of retina displays we need to use high resolution images whenever possible. However using always the highest resolution images will have an impact on performance as more bytes will need to be sent over the wire.

To overcome this problem, we can use responsive images, as specified in HTML5. It requires making available different resolution files of the same image to the browser and let it decide which image is best, using the html attribute `srcset` and optionally `sizes`, for instance:

```
<div responsive-background-image>
  <img src="/images/test-1600.jpg"
    sizes="
      (min-width: 768px) 50vw,
      (min-width: 1024px) 66vw,
      100vw"
    srcset="
      /images/test-400.jpg 400w,
      /images/test-800.jpg 800w,
      /images/test-1200.jpg 1200w">
</div>
```

It is important to note that browsers which don't support HTML5's `srcset` (i.e. IE11) will ignore it and use `src` instead. If we really need to support IE11 and we want to provide this feature for performance reasons, we can use a JavaScript polyfill, e.g. Picturefill (link in the references).

For icons, I would also opt to use SVGs and icon fonts where possible, as they render very crisply regardless of resolution.

References

- https://css-tricks.com/responsive-images-youre-just-changing-resolutions-use-srcset/
- http://scottjehl.github.io/picturefill/
- https://aclaes.com/responsive-background-images-with-srcset-and-sizes/

Is there any reason you'd want to use translate() instead of absolute positioning, or vice-versa? And why?

`translate()` is a value of CSS `transform`. Changing `transform` or `opacity` does not trigger browser reflow or repaint but does trigger compositions; whereas changing the absolute positioning triggers `reflow`. `transform` causes the browser to create a GPU layer for the element but changing absolute positioning properties uses the CPU. Hence `translate()` is more efficient and will result in shorter paint times for smoother animations.

When using `translate()`, the element still occupies its original space (sort of like `position: relative`), unlike in changing the absolute positioning.

References

- https://www.paulirish.com/2012/why-moving-elements-with-translate-is-better-than-posabs-topleft/

The translate() function can move the position of an element on the z-axis.

- False

Do you subscribe to any particular CSS structure? (SMACSS, OOCSS)

What is browser Compatibility?

Tell what each of these tags do, if there are alternatives, which are preferable, why?

```
<em>
<b>
<abbr>
<nav>
<i>
<j>
<link>
<strong>
<article>
```

What is At-Rule

- An at-rule is an instruction given in a CSS document using the `@` character. An at-rule could have a declaration block or a simple string of text. The example below has two different at-rules:

```css
@import url(secondary.css);

@media print {
  #container {
    width: 500px;
  }
}
```

The at-rule is not just the `@media` or `@import` part at the beginning; the entire instruction comprises the complete at-rule.

What is Combinator selector

- A combinator is the character in a selector that connects two selectors together. There are four types of combinators. These four combinators help create descendant selectors (with a space character), child selectors (with the `>` character), adjacent sibling selectors (with the `+` character), and general sibling selectors (with the `~` character). To dispel any confusion, here are those four combinators in use:

```
/*  In all 4 examples */
/*  whatever appears between "div" and "p" is a combinator */
/*  in the first example, the combinator is a space character */
div p {
  color: #222;
}

div>p {
  color: #333;
}

div+p {
  color: #444;
}

div~p {
  color: #555;
}
```

What is RuleSet?

- A rule set is a single section of CSS including the selector, the curly braces, and the different lines with properties and values. The code in the example below comprises one rule set:

```
/*  the rule set starts with the line below */
body {
  font-family: Arial, sans-serif;
  color: #555;
  font-size: 14px;
}
/*  ends with the closing curly brace above */
```

seeAndTell

```
https://www.thatjsdude.com/interview/css.html#seeAndTell
```

Provide CSS so that three lines are displayed red, then green, then blue (2):

```
<div class="rgb">
    <span>Red</span>
    <span>Green<b>Blue</b></span>
</div>
```

Provided following HTML and CSS, what will be displayed (1)?

```
<div class="red">Hi, I'm Blue</div>

div.red { color: green; }
div { font-weight: bold; }
.red { color: yellow; font-weight: normal; }
```

What is Declaration

- The set of property names and values like: background: red;

What is Declaration Block

- A declaration block is the section of CSS where the property/value pairs appear. In the example below, everything found between the curly braces (not including the comments) is a declaration block:

```
body {
  font-family: Arial, sans-serif; /*  starts with this line */
  color: #555;
  font-size: 14px;
  line-height: 20px; /*  ends here, before the closing curly brace */
}
```

What is Universal Selector

- The universal selector matches any element within the context in which it's placed in a selector. In the example below, the `*` character is the universal selector:

```
/*  the asterisk character is the universal selector */
.navigation *ul {
  width: 100px;
  float: left;
}
```

- So, any element that appears as a child descendant of the unordered list element inside an element that has a class of "navigation" will receive the declared styles.
- Universal selectors are generally discouraged for performance reasons.

How can the gap under the image be removed?

- As images being inline elements are treated same as texts, so there is a gap left, which can be removed by:

```
img { display: block ; }
```

How comments can be added in CSS?

- The comments in CSS can be added with /* and */.

Define Attribute Selector ?

- It is defined by a set of elements, value and its parts.

Define property?

- A style, that helps in influencing CSS. E.g. FONT. They have corresponding values or properties within them, like FONT has different style like bold, italic etc.

What is graceful degradation?

- In case the component fails, it will continue to work properly in the presence of a graceful degradation. The latest browser application is used when a webpage is designed. As it is not available to everyone, there is a basic functionality, which enables its use to a wider audience. In case the image is unavailable for viewing, text is shown with the alt tag.

What is progressive enhancement?

- It’s an alternative to graceful degradation, which concentrates on the matter of the web. The functionality is same, but it provides an extra edge to users having the latest bandwidth. It has been into prominent use recently with mobile internet connections expanding their base.

What is progressive rendering?

- Progressive rendering is the name given to techniques used to improve the performance of a webpage (in particular, improve perceived load time) to render content for display as quickly as possible.
- It used to be much more prevalent in the days before broadband internet but it is still used in modern development as mobile data connections are becoming increasingly popular (and unreliable)!

http://stackoverflow.com/questions/33651166/what-is-progressive-rendering

- Progressive rendering is the name given to techniques used to improve the performance of a webpage (in particular, improve perceived load time) to render content for display as quickly as possible.
- It used to be much more prevalent in the days before broadband internet but it is still used in modern development as mobile data connections are becoming increasingly popular (and unreliable)!
- Examples of such techniques:
    - Lazy loading of images - Images on the page are not loaded all at once. JavaScript will be used to load an image when the user scrolls into the part of the page that displays the image.
    - Prioritizing visible content (or above-the-fold rendering) - Include only the minimum CSS/content/scripts necessary for the amount of page that would be rendered in the users browser first to display as quickly as possible, you can then use deferred scripts or listen for the DOMContentLoaded/load event to load in other resources and content.
    - Async HTML fragments - Flushing parts of the HTML to the browser as the page is constructed on the back end.

What is feature detection (vs browser detection)?

What is an `iframe` and how it works?

- An `iframe` is an HTML document which can be embedded inside another HTML page.
- The IFrame HTML element is often used to insert content from another source, such as an advertisement, into a Web page.

```
<iframe src="https://github.com" height="300px" width="300px"></iframe>
```

What is mobile-first? Can you explain the difference between coding a website to be responsive versus using a mobile-first strategy

Note that these two 2 approaches are not exclusive.

Making a website responsive means the some elements will respond by adapting its size or other functionality according to the device's screen size, typically the viewport width, through CSS media queries, for example, making the font size smaller on smaller devices.

```
@media (min-width: 601px) {
  .my-class {
    font-size: 24px;
  }
}
@media (max-width: 600px) {
  .my-class {
    font-size: 12px;
  }
}
```

A mobile-first strategy is also responsive, however it agrees we should default and define all the styles for mobile devices, and only add specific responsive rules to other devices later. Following the previous example:

```
.my-class {
  font-size: 12px;
}

@media (min-width: 600px) {
  .my-class {
    font-size: 24px;
  }
}
```

A mobile-first strategy has 2 main advantages:

- It's more performant on mobile devices, since all the rules applied for them don't have to be validated against any media queries.
- It forces to write cleaner code in respect to responsive CSS rules.

How is responsive design different from adaptive design?

Both responsive and adaptive design attempt to optimize the user experience across different devices, adjusting for different viewport sizes, resolutions, usage contexts, control mechanisms, and so on.

Responsive design works on the principle of flexibility - a single fluid website that can look good on any device. Responsive websites use media queries, flexible grids, and responsive images to create a user experience that flexes and changes based on a multitude of factors. Like a single ball growing or shrinking to fit through several different hoops.

Adaptive design is more like the modern definition of progressive enhancement. Instead of one flexible design, adaptive design detects the device and other features and then provides the appropriate feature and layout based on a predefined set of viewport sizes and other characteristics. The site detects the type of device used and delivers the pre-set layout for that device. Instead of a single ball going through several different-sized hoops, you'd have several different balls to use depending on the hoop size.

References

- https://developer.mozilla.org/en-US/docs/Archive/Apps/Design/UI_layout_basics/Responsive_design_versus_adaptive_design
- http://mediumwell.com/responsive-adaptive-mobile/
- https://css-tricks.com/the-difference-between-responsive-and-adaptive-design/

What is accessibility/usability? What experience do you have with web accessibility?

What does Accessibility (a11y) mean?

Accessibility (a11y) is a measure of a computer system's accessibility is to all people, including those with disabilities or impairments. It concerns both software and hardware and how they are configured in order to enable a disabled or impaired person to use that computer system successfully.

Accessibility is also known as assistive technology.

Source

```
https://www.techopedia.com/definition/10165/accessibility-a11y
```

What is accessibility & ARIA role means in a web application?

Accessibility means letting the user navigate and interact with a website by any means. Let it be a keyboard, a mouse or people with disabilities or impairments.

- ARIA stands for Accessible Rich Internet Application.
- ARIA is a spec from W3C.
- ARIA is created to improve accessibility in web applications.
- ARIA provides a extra information to screen readers via HTML attributes.
- ARIA doesn't affect how an element is rendered in a browser.

What is SEO

What is UI/UX

- UI—or User Interface—is how a product or website is laid out and how you interact with it: Where the buttons are, how big the fonts are, and how menus are organized are all elements of UI. But UX—or User Experience—is how you feel about using a product or a website. So, your love for the way the new Apple Watch looks or your excitement that there's finally a tablet-sized iPhone to watch those Corgi videos you're obsessed with are reflections of UX. So the new look of the Facebook news feed involves a change to UI, and the way you navigate that new page is the UX.

What is semantic

Semantic HTML, or "semantically-correct HTML", is HTML where the tags used to structure content are selected and applied appropriately to the meaning of the content.

For example, (for bold), and (for italic) should never be used, because they're to do with formatting, not with the meaning or structure of the content. Instead, use the replacements and (meaning emphasis), which by default will turn text bold and italic (but don't have to do so in all browsers), while adding meaning to the structure of the content

Why you would like to use semantic tag. Why Semantic HTML is Important?

- Search Engine Optimization, accessibility, repurposing, light code.
- Many visually impaired person rely on browser speech and semantic tag helps to interpret page content clearly.
- Search engine needs to understand page content to rank and semantic tag helps.
- semantic code aids accessibility. Specially, many people whose eyes are not good rely on speech browsers to read pages to them. These programs cannot interpret pages very well unless they are clearly explained.
- Help Search engines to better understand pages. Search engine need to understand what your content is about when rank you properly on search engines. Semantic code tends to improve your placement on search engines, as it is easier for the "search engine spiders" to understand.
- It's easier to read and edit, which saves time and money during maintenance.

How Can I Get Indexed Better by Search Engines?

What does "semantically correct" mean?

https://stackoverflow.com/questions/1294493/what-does-semantically-correct-mean/1294512#1294512

What is typography

What are web standards

What is valid HTML i CSS? (Who determines standards and criteria)

Which of listed CSS properties are inherited?

- position
- font-size
- color
- background-color

Enlist image types used in web, their properties and usage

Which property is used to change the face of a font?

- The font-family property is used to change the face of a font.

Which property is used to make a font italic or oblique?

- The font-style property is used to make a font italic or oblique.

Which property is used to create a small-caps effect?

- The font-variant property is used to create a small-caps effect.

Which property is used to increase or decrease how bold or light a font appears?

- The font-weight property is used to increase or decrease how bold or light a font appears.

Which property is used to add or subtract space between the letters that make up a word?

- The letter-spacing property is used to add or subtract space between the letters that make up a word.

Which property is used to add or subtract space between the words of a sentence?

- The word-spacing property is used to add or subtract space between the words of a sentence.

Which property is used to indent the text of a paragraph?

- The text-indent property is used to indent the text of a paragraph.

Which property is used to align the text of a document?

- The text-align property is used to align the text of a document.
- `text-align: center` will do the horizontal alignment but vertical-align: middle will not work here. there are couple of different ways to solve this problem and one of them are positioning. You make the parent as relative position and child as absolute positioning. And then define all position parameter as sero and width 50% and height 30% (sounds messy look at the example and read ref)

Which property is used to underline, overline, and strikethrough text?

- The text-decoration property is used to underline, overline, and strikethrough text.

Which property is used to capitalize text or convert text to uppercase or lowercase letters?

- The text-transform property is used to capitalize text or convert text to uppercase or lowercase letters.

Which property allows you to control the shape or appearance of the marker of a list?

- The list-style-type allows you to control the shape or appearance of the marker.

How do I restore the default value of a property?

- Initially CSS didn't provide a "default" keyword and the only way to restore the default value of a property is to explicitly re-declare that property This has changed with CSS 2; the keyword 'initial' is now a valid value for a CSS property. It resets it to its default value, which is defined in the CSS specification of the given property.

How do I assign multiple classes to an element?

- HTML elements can be assigned multiple classes by listing the classes in the class attribute, with a blank space to separate them.

```
<style type="text/css">
.news { background: black; color: white; }
.today { font-weight: bold; }
</style>

<div class="news today">
... content of today's news ...
</div>
```

What are preferred rules for specificity?

What is specificity? How do u calculate specificity?

- a process of determining which css rule will be applied to an element. it actually determines which rules will take precedence.
- inline style usually wins then ID then class value (or pseudo-class or attribute selector), universal selector (*) has no specificity.

What is selector specificity (selector importance) and how it works? How do u calculate specificity?

- https://developer.mozilla.org/en/docs/Web/CSS/Specificity
- https://www.smashingmagazine.com/2007/07/css-specificity-things-you-should-know/
- https://github.com/TIY-Austin-Front-End-Engineering/css-specificity

ID selectors have a higher specificity than attribute selectors.

```
/*wins*/
a#a-02 { background-image : url(n.gif); }
a[id="a-02"] { background-image : url(n.png); }
```

Contextual selectors are more specific than a single element selector. The embedded style sheet is closer to the element to be styled. The last rule defined overrides any previous, conflicting rules.

```
p { color: red; background: yellow }
p { color: green } // wins
```

A class selector beats any number of element selectors.

```
.introduction {} //wins
html body div div h2 p {}
```

The universal selector has a specificity of 0, 0, 0, 0.

What is CSS selector specificity and how does it work?

What do you know about transition?

- transition allows to add an effect while changing from one style to another. You can set the which property you want to transition, duration, how you want to transit (linear, ease, ease-in, ease-out, cubic-bezier) and delay when transition will start. you can transition more than one property by comma separation

What are the different css filter you can use?

```
https://developer.mozilla.org/en/docs/Web/CSS/filter
```

- css filter allows u to render DOM element, image, or video. u can choose from: grayscale, blur, opacity, brightness, contrast.

Enlist the various fonts properties?

- font-style
- font-variant
- font-weight
- font-size/line-height
- font-family
- caption
- icon

Enlist the position / display / overflow properties and what are defaults

Enlist the various CSS background properties.

Image paths in CSS vs HTML

What is transition & transform?

What is used flexbox / transforms / transitions / animations?

What exposure do you have to CSS3 animations and transitions

Can you export contents from PSD / Sketch?

What kind of challenges do you run into formatting a site for Mobile devices?

Describe Floats and Flexbox and how each works. Why Flexbox is a better solution for web layouts?

- `Floats` specifies that an element should be taken from the normal flow and placed along the left or right side of its container, where text and inline elements will wrap around it.

- `Flexbox` consists of flex containers and flex items. Flex containers wrap a set of flex items and define how they are laid out. Flex items has properites that define how they interact with sibling flex elements and can also be flex containers themselves.

What font units are available in CSS?

What is the difference between `em` and `rem` units?

Both em and rem units are based on the font-size CSS property. The only difference is where they inherit their values from.

- `em` units inherit their value from the font-size of the parent element
- `rem` units inherit their value from the font-size of the root element (html)

In most browsers, the font-size of the root element is set to 16px by default.

What are Benefits of using `em` and `rem` units

What is the difference between padding and margins?

What is the difference between `span` and `div` ?

`div` is a block element and `span` is inline element.

It is illegal to put block element inside inline element. div can have a p tag and a p tag can have a span. However, span can't have a div or p tag inside.

What does !important mean in CSS?

- It overrides the cascade and gives the style rule the highest precedence.

What is cross-browser compatibility?

What are optional closing tag? and why would u use it?

`p, li, td, tr, th, html, body` , etc. you don't have to provide end tag. Whenever browser hits a new tag it automatically ends the previous tag. However, you have to be careful to escape it.

What is a self closing tag?

In HTML5 it is not strictly necessary to close certain HTML tags. The tags that aren't required to have specific closing tags are called "self closing" tags.

An example of a self closing tag is something like a line break (`<br />`) or the meta tag (`<meta>`). This means that the following are both acceptable:

```
<meta charset="UTF-8">
...
<meta charset="UTF-8" />
```

How do you test cross-browser compatibility of your site?

How can you make a site responsive?

The Difference Between Responsive and Adaptive Design

```
https://css-tricks.com/the-difference-between-responsive-and-adaptive-design/
```

What do you know about text shadows, box shadows?

How can u highlight text in html?

use mark element.

```
<p>Some part of this paragraph is <mark>highlighted</mark> by using mark element.</p>
```

What is the difference between CSS variables and preprocessor variables?

```
https://css-tricks.com/difference-between-types-of-css-variables/
```

The Difference Between :nth-child and :nth-of-type

```
https://css-tricks.com/the-difference-between-nth-child-and-nth-of-type/
```

What selects ul[class] > li:nth-child(2n+1)

What is the difference between “word-break: break-all” versus “word-wrap: break-word” in CSS

How is word-wrap different from white-space ?

```
https://css-tricks.com/almanac/properties/w/whitespace/
```

What is `<pre>` ?

How is `<b>` different from `<strong>` ?

What does minification do?

What are some ways to make websites faster? Name as many different techniques as you can.

Does the following trigger http request at the time of page load?

```
<img src="mypic.jpg" style="visibility: hidden" alt="My photo">
```

```
<div style="display: none;">
    <img src="mypic.jpg" alt="My photo">
</div>
```

- Yes

Does style2.css have to be downloaded and parsed before Paragraph 1 is rendered on the page?

```
<head>
    <link href="style1.css" rel="stylesheet">
</head>
<body>
    <p>Paragraph 1</p>
    <p>Paragraph 2</p>
    <link href="style2.css" rel="stylesheet">
</body>
```

- Yes

Does style1.css have to be downloaded and parsed before style2.css can be fetched?

```
<head>
    <link href="style1.css" rel="stylesheet">
    <link href="style2.css" rel="stylesheet">
</head>
```

- No

Can u apply css rule to a part of html document?

yes. by using "scopped" in the style tag.

How do you test the performance of your code and/or web pages?

What are some rules for writing efficient CSS

Firstly, understand that browsers match selectors from rightmost (key selector) to left. Browsers filter out elements in the DOM according to the key selector and traverse up its parent elements to determine matches. The shorter the length of the selector chain, the faster the browser can determine if that element matches the selector. Hence avoid key selectors that are tag and universal selectors. They match a large number of elements and browsers will have to do more work in determining if the parents do match.

BEM (Block Element Modifier) methodology recommends that everything has a single class, and, where you need hierarchy, that gets baked into the name of the class as well, this naturally makes the selector efficient and easy to override.

Be aware of which CSS properties trigger reflow, repaint, and compositing. Avoid writing styles that change the layout (trigger reflow) where possible.

References

- https://developers.google.com/web/fundamentals/performance/rendering/
- https://csstriggers.com/

Sass

SASS basic concepts

- Variables: Variables in SASS start with $ sign
- Nesting: CSS lacks visual hierarchy while working with child selectors. You have to write selectors and their combinations in separate lines. Nesting provides a visual hierarchy as in the HTML and increases the readability.

- mixins: mixins are used to include a bunch of properties or group declarations together. It allows for the easy reuse of blocks of code. Use include to
- Inheritance: extends are useful for sharing a generic definition with selectors rather than copying it in.
- If/Else Statements and loops
- import: separating your codes in small pieces is helpful for expressing your declarations and increasing maintainability and control over the codebase.
- Math operations: can be used for standard arithmetic or unit conversions.

What are you favorite features in SASS?

Explain nested selectors in Sass

How do you refer to a parent selector in the SASS?

Explain what Sass Maps are and how they are use?

Explain the @import function and how it is used

Explain the @include, @mixin, @function functions and how they are used. What is %placeholder

- `@mixin` A mixin lets you make groups of CSS declarations that you want to reuse throughout your site

```
@mixin border-radius($radius) {
  -webkit-border-radius: $radius;
     -moz-border-radius: $radius;
      -ms-border-radius: $radius;
          border-radius: $radius;
}


.box { @include border-radius(10px); }
```

- `@extend` directive provides a simple way to allow a selector to inherit/extend the styles of another one.

```
.message {
  border: 1px solid #ccc;
  padding: 10px;
  color: #333;
}

.success {
  @extend .message;
  border-color: green;
}

.error {
  @extend .message;
  border-color: red;
}
```

`%placeholder` are classes that aren’t output when your SCSS is compiled

```
%awesome {
    width: 100%;
    height: 100%;
}
body {
    @extend %awesome;
}
p {
    @extend %awesome;
}
```

```
/* Output */
body, p {
    width: 100%;
    height: 100%;
}
```

What are some considerations in selecting font sizes? // This question is focused on accessibility

As an organisation we aim to meet WCAG AA standards for accessibility. What experience do you have with developing accessible websites?

How do you approach developing an accessible website?

What are some of the common accessibility pitfalls when developing a website? How would you go about fixing them?

HTML

What does a DOCTYPE do?

- DOCTYPE is an abbreviation for `DOCument TYPE` .
  A DOCTYPE is always associated to a `DTD` for `Document Type Definition` .

  A DTD defines how documents of a certain type should be structured (i.e. a `button` can contain a `span` but not a `div`), whereas a DOCTYPE declares what DTD a document supposedly respects (i.e. this document respects the HTML DTD).

  For webpages, the DOCTYPE declaration is required. It is used to tell user agents what version of the HTML specifications your document respects.
  Once a user agent has recognized a correct DOCTYPE, it will trigger the `no-quirks mode` matching this DOCTYPE for reading the document.
  If a user agent doesn't recognize a correct DOCTYPE, it will trigger the `quirks mode` .

  The DOCTYPE declaration for the HTML5 standards is `<!DOCTYPE html>` .

- it controls whether the browsers uses "standards" or "quirks" mode to render the document.

What happens when DOCTYPE is not given?

- The web page is rendered in quirks mode. The web browsers engines use quirks mode to support older browsers which does not follow the W3C spec.

  In quirks mode CSS class and id names are case insensitive. In standards mode they are case sensitive.

What's the difference between standards mode and quirks mode?

- Quirks mode was to support websites built before standards became widely implemented.

- quirks mode in browser allows u to render page for as old browsers. This is for backward compatibility.

What's the difference between HTML and XHTML?

The Extensible Hypertext Markup Language, or XHTML, has two important notes for front end developers. 1) It needs to be well formed, meaning all elements need to be closed and nested correctly or you will return errors. 2) Since it is more strict than HTML is requires less pre-processing by the browser, which may improve your sites performance.

Why html5 doctype does not have a DTD definition?

- HTML5 is no longer based on SGML (Standard Generalized Markup Language) which actually requires a DTD for parsing/serializing, so we don't require a DTD anymore.

What does the lang attribute in html do?

- Helps in styling pages bu using them in css :lang() pseudo class Spelling and grammar checkers Languade detection by search engines

What is desktop first and mobile first design approach

- Desktop first : General selectors and styles designed to make the site look good on DESKTOP screens defined globally. But they affect all devices, and must be overridden by max-width media queries targeting minimum screen size
- Mobile First : General selectors and styles designed to make the site look good on small MOBILE screens go here. But they affect all devices, and must be overridden by min-width media queries targeting maximum scrren size

  In desktop first approach the media queries will be written with respect to max-width whereas in mobile first approach media queries will be written with respect to min-width

Are there any problems with serving pages as application/xhtml+xml?

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

- The question is a little vague, I will assume that it is asking about the most common case, which is how to serve a page with content available in multiple languages, but the content within the page should be displayed only in one consistent language.
- When an HTTP request is made to a server, the requesting user agent usually sends information about language preferences, such as in the `Accept-Language` header. The server can then use this information to return a version of the document in the appropriate language if such an alternative is available. The returned HTML document should also declare the `lang` attribute in the `<html>` tag, such as `<html lang="en">...</html>` .
- In the back end, the HTML markup will contain `i18n` placeholders and content for the specific language stored in YML or JSON formats. The server then dynamically generates the HTML page with content in that particular language, usually with the help of a back end framework.

What kind of things must you be wary of when design or developing for multilingual sites?

- Use `lang` attribute in your HTML.
- Directing users to their native `language` Allow a user to change his country/language easily without hassle.
- Text in images is not a scalable approach Placing text in an image is still a popular way to get good-looking, non-system fonts to display on any computer. However, to translate image text, each string of text will need to have a separate image created for each language. Anything more than a handful of replacements like this can quickly get out of control.
- Restrictive words/sentence `length` Some content can be longer when written in another language. Be wary of layout or overflow issues in the design. It's best to avoid designing where the amount of text would make or break a design. Character counts come into play with things like headlines, labels, and buttons. They are less of an issue with free-flowing text such as body text or comments.
- Be mindful of how colors are `perceived` Colors are perceived differently across languages and cultures. The design should use color appropriately.
- Formatting dates and `currencies` Calendar dates are sometimes presented in different ways. Eg. "May 31, 2012" in the U.S. vs. "31 May 2012" in parts of Europe.

- Do not concatenate translated `strings` Do not do anything like `"The date today is " + date` . It will break in languages with different word order. Use a template string with parameters substitution for each language instead. For example, look at the following two sentences in English and Chinese respectively: `I will travel on {% date %}` and `{% date %} 我会出发` . Note that the position of the variable is different due to grammar rules of the language.
- Language reading `direction` In English, we read from left-to-right, top-to-bottom, in traditional Japanese, text is read up-to-down, right-to-left.

What are `data-` attributes good for?

- The HTML5 data attribute lets you assign custom data to an element. When we want to store more information/data about the element when no suitable HTML5 element or attribute exists

Consider HTML5 as an open web platform. What are the building blocks of HTML5?

- more semantic text markup
- new form elements
- new video and audio elements
- javascript API
- canvas and SVG
- geolocation API
- new data storage

Describe the difference between a `cookie` , `sessionStorage` and `localStorage` .

- `cookie` : A text file saved on the users computer to store and retrieve data
- `sessionStorage` : Is memory space in a browser to save temporary data until the window or tab is closed.
- `localStorage` : Like cookie, where data can be saved and retrieved after browser sessions, but stored in memory like sessionStorage. Data is stored as plain key value pairs and can be stored as Json objects.

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

- Normal Execution: script: Parsing of the HTML code pauses while the script is executing. For slow servers and heavy scripts this means that displaying the webpage will be delayed.
- Deferred Execution: script defer Delays script execution until the HTML parser has finished. A positive effect of this attribute is that the DOM will be available for your script. However, since not every browser supports defer yet, can't rely on it.
- Asynchronous Execution: script async HTML parsing may continue and the script will be executed as soon as it's ready.

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

Placing `<link>` s in the `<head>`

Putting `<link>` s in the head is part of proper specification in building an optimized website. When a page first loads, HTML and CSS are being parsed simultaneously; HTML creates the DOM (Document Object Model) and CSS creates the CSSOM (CSS Object Model). Both are needed to create the visuals in a website, allowing for a quick "first meaningful paint" timing. This progressive rendering is a category optimization sites are measured in their performance scores. Putting stylesheets near the bottom of the document is what prohibits progressive rendering in many browsers. Some browsers block rendering to avoid having to repaint elements of the page if their styles change. The user is then stuck viewing a blank white page. Other times there can be flashes of unstyled content (FOUC), which can shows a webpage with no styling applied.

Placing `<script>` s just before `</body>`

`<script>` s block HTML parsing while they are being downloaded and executed. Placing the scripts at the bottom will allow the HTML to be parsed and displayed to the user first.

An exception for positioning of `<script>` s at the bottom is when your script contains `document.write()` , but these days it's not a good practice to use `document.write()` . Also, placing `<script>` s at the bottom means that the browser cannot start downloading the scripts until the entire document is parsed. This ensures your code that needs to manipulate DOM elements will not throw and error and halt the entire script. If you need to put `<script>` in the `<head>` , use the `defer` attribute, which will achieve the same effect of downloading and running the script only after the HTML is parsed.

References

- https://developer.yahoo.com/performance/rules.html#css_top
- https://www.techrepublic.com/blog/web-designer/how-to-prevent-flash-of-unstyled-content-on-your-websites/
- https://developers.google.com/web/fundamentals/performance/critical-rendering-path/

What is progressive rendering?

It is rendering the data as it's being downloaded. This is particularly useful on documents that have tons of text. You can see it on a page that has a lot of text – and where the scrollbar will get shorter in length as more data comes in – increasing the vertical size of the document – yet, it would display the downloaded text immediately. As more data came down the pipe – the page would get longer. This didn't rely on the closing body or html tag – and it certainly wouldn't render the entire page on the server – then download – which is a standard complaint about modern frameworks. But there is a technique called "Flushing the Buffer" that can be implemented on the server. I don't know that much about the technique, but found a few resources discussing it.

What is Critical Rendering Path

- Constructing the DOM Tree
- Constructing the CSSOM Tree
- Running JavaScript - parser blocking resource
- Creating the Render Tree
- Generating the Layout
- Painting

SSR VS CSR

- We are using server side rendering for two reasons:
    - performance benefit for our customers
    - Consistent SEO performance
- The main difference is that for SSR your server's response to the browser is the HTML of your page that is ready to be rendered, while for CSR the browser gets a pretty empty document with links to your javascript. That means for SSR your browser will start rendering the HTML from your server without having to wait for all the JavaScript to be downloaded and executed.
- for SSR, the user can start viewing the page while all of that is happening. For the CSR world, you need to wait for all of the above to happen and then have the virtual dom moved to the browser dom for the page to be viewable.

Have you used different HTML templating languages before? what did you like about them?

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

https://github.com/FAQGURU/FAQGURU/blob/master/topics/en/html5.md#why-you-would-use-a-srcset-attribute-in-an-image-tag-explain-the-process-the-browser-uses-when-evaluating-the-content-of-this-attribute

Why does invalid HTML work?

What is XHTML?

- https://www.thoughtco.com/what-is-xhtml-3464702

What is a Favicon? Where are favicons displayed?

Are HTML 5 tags case sensitive?

Why must the ID attribute be unique on each page?

What is a blockquote?

What is a meta tag?

How many cookies can you use on one website?

How big can a web cookie be?

What is the difference between Canvas and SVG graphics?

https://www.sitepoint.com/how-to-choose-between-canvas-and-svg/

What is the difference between a `<span>` and a `<div>` ?

- `<div>` is a block level element which means it will render it on it's own line with a width of a 100% of the parent element.
- `<span>` is an inline element which means it will render on the same line as the previous element, if it is also an inline element, and it's width will be determined by it's content.

Name 5 common block-level and inline HTML elements.

- block elements `<h1>, <p>, <ul>, <ol>, <li>,`
- inline elements `<span>, <a>, <strong>, <i>, <img>`

What are semantic and non-semantic elements?

- A semantic element clearly describes its meaning to both the browser and the developer.
- non-semantic elements: `<div>` and `<span>` Tells nothing about its content. semantic elements: `<form>, <table>, and <article>` Clearly defines its content.

What is the purpose of `main` element?

The HTML `<main>` element represents the dominant content of the of a document, portion of a document or application. The main content area consists of content that is directly related to or expands upon the central topic of a document, or the central functionality of an application. One important facet of `<main>` is that it can only be used once per page.

`<main>` doesn't contribute to the document's outline; that is, unlike elements such as `<body>`, headings such as `<h2>`, and such, `<main>` doesn't affect the DOM's concept of the structure of the page. It's strictly informative.

Why to use HTML5 semantic tags?

As their name says, this is for semantic purposes only. It's for improving the automated processing of documents. Automated processing happens more often than you realize - each website ranking from search engines is derived from automated processing of all the website out there.

```
// machine: okay, this structure looks like it might be a navigation element?
<div class="some-meaningless-class"><ul><li><a href="internal_link">...</div>

// machine: ah, a navigation element!
<nav class="some-meaningless-class"><ul><li><a>...</nav>
```

Define semantic markup. What are the semantic meanings for `<section>, <article>, <aside>, <nav>, <header>, <footer>` and when/how should each be used in structuring html markup?

- `<header>` is used to contain introductory and navigational information about a section of the page. This can include the section heading, the author's name, time and date of publication, table of contents, or other navigational information.
- `<article>` is meant to house a self-contained composition that can logically be independently recreated outside of the page without losing it's meaining. Individual blog posts or news stories are good examples.
- `<section>` is a flexible container for holding content that shares a common informational theme or purpose.
- `<footer>` is used to hold information that should appear at the end of a section of content and contain additional information about the section. Author's name, copyright information, and related links are typical examples of such content.

When should you use `section`, `div` or `article` ?

- `<section>`, group of content inside is related to a single theme, and should appear as an entry in an outline of the page. It's a chunk of related content, like a subsection of a long article, a major part of the page (eg the news section on the homepage), or a page in a webapp's tabbed interface. A section normally has a heading (title) and maybe a footer too.
- `<article>`, represents a complete, or self-contained, composition in a document, page, application, or site and that is, in principle, independently distributable or reusable, e.g. in syndication. This could be a forum post, a magazine or newspaper article, a blog entry, a user-submitted comment, an interactive widget or gadget, or any other independent item of content.
- `<div>`, on the other hand, does not convey any meaning, aside from any found in its class, lang and title attributes.

What is the difference between (`<i>` and `<em>`) and (`<b>` , `<strong>`) and how will screen-readers react the to each tag?

What is the purpose of aria attributes?

Explain what inline styles and javascript are and when they are appropriate to use.

When is it acceptable to use font tags?

What is Character Encoding?

To display an HTML page correctly, a web browser must know which character set (character encoding) to use. This is specified in the tag:

```
<meta charset="UTF-8">
```

What is the purpose of meta tags?

- Meta tags always go inside head tag of the HTML page
- Meta tags is always passed as name/value pairs
- Meta tags are not displayed on the page but intended for the browser
- Meta tags can contain information about character encoding, description, title of the document etc

```
<!DOCTYPE html>
<html>
<head>
  <meta name="description" content="I am a web page with description">
  <title>Home Page</title>
</head>
<body>

</body>
</html>
```

What is the purpose of Charset meta tag?

What does async and defer refer in script tag ? Describe the difference between `<script>` , `<script async>` and `<script defer>`

- Async: Downloads the script file during HTML parsing and will pause the HTML parser to execute it when it has finished downloading.
- Defer: Defer downloads the script file during HTML parsing and will only execute it after the HTML parser has completed. Not all browsers support this.
- the async attribute is used to indicate to the browser that the script file can be executed asynchronously. The HTML parser does not need to pause at the point it reaches the script tag to fetch and execute, the execution can happen whenever the script becomes ready after being fetched in parallel with the document parsing.
- The defer attribute tells the browser to only execute the script file once the HTML document has been fully parsed.

Experience with website optimization?

How to handle form validation?

What is viewport?

What is the difference between cookies, sessionStorage and localStorage?

Where should we place Javascript `files` top OR bottom. What are the advantages and disadvantages of either method. Where would you use these (Google webfonts you need to place on top + jQuery down or top)

Behavioral

What are your biggest strengths? Weaknesses?

What is your greatest achievement?

Tell me about 3 times you have failed.

```
// Almost anyone can come up with 1 or 2, but it can be hard to think of 3.
// Be sure to wait and be quiet while the candidate thinks and ponders the answer
// I often feel like the most insightful answers come towards the end of this question.
```

How did you prepare for this interview?

What are some of your hobbies or passions outside of work? How did you first get into them?

What is your availability?

Do you consider yourself weird? Why or why not?

Are there any questions that we didn’t ask you that we should have?

Why did you choose to learn FE?

How did you learn to code?

What sort of project are you looking for?

Give me an example of a goal you set for yourself and how you achieved it.

If money weren’t an issue and you had to work on a project for 3 months, what would you create?

Where do you see yourself/ What do you want to do in 3/5 years? What kind of environment? How would this job fit into your plan and help you toward your goals?

If you could master one technology this year, what would it be?

What was your greatest challenge in FE?

What was your favorite part about FE?

What excites or interests you about coding?

What has been the hardest part about learning to code so far?

What did you learn yesterday/this week?

What are 3 big contributors to your success? Who inspires you in the front-end community?

What is the last technical book you read?

What sort of websites or blogs do you read regularly? What do you like about them?

Do you use Git, GitHub?

Tell me about a time you improved a tool, task, or project you were working on. What was the circumstances? Why did you do it? Do you have any other examples?

Do you have any side projects? What kind?

What's the coolest thing/project you've ever coded, what are you most proud of?

What is a recent technical challenge you experienced and how did you solve it?

Talk about your preferred development environment.

What tools you use in regular workflow?

What tools do you use for your day to day work? Language, IDEs/editors, version control, build systems, provisioning, etc.?

Which version control systems are you familiar with?

How do you stay on up-to-date with current web trends/innovations, latest front-end technologies and best practices? how do you decide what to invest time into and what not to?

What are some things you like about the developer tools you use?

What is a recent challenges you experienced and how did you solved those?

What was the last new technology or tool you learned? Where did you learn about it? Have you used it since?

When was the last time you got something wrong? How did you know? What did you learn from it?

Can you describe your workflow when you create a web page?

If you have 5 different stylesheets, how would you best integrate them into the site?

Break them up onto to different CDN servers to leverage domain sharding.

Employ the "new old" technique of adding "above the fold" css inline in the head of your document – reducing http requests and improve perceived performance.

Using SASS I would break up my files into related chunks – all of which are concatenated and compressed using compass, sass with gulp or grunt in your build process.

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

- Graceful degradation is when you initially serve the best possible user experience, with all modern functionality, but use feature detection to "gracefully degrade" parts of your application with a fallback or polyfill.
- Progressive enhancement ensures a page works at the lowest expected abilities of browsers. So if you have a JavaScript web application that enhances a persons ability to send information to a database with features like ajax – at the very least you need to provide the ability for a person to send that same information without JavaScript enabled. In this case a simple form with full-page refresh will do what you need.

How would you optimize a website's assets/resources?

Concatenate and compress CSS, JavaScript and HTML files wherever possible, configure your server to deliver a Gzip files, cache resources, set longer expirations dates on http headers of resources you don't expect to change often – such as a logo. Images can be some of the heaviest files we deliver, so compress wisely. Soon the picture element will be implemented across browsers, so we can optimize the delivery of image content. Also in the near future consider using WebP format for images – it is quite smaller in size than JPEG and PNG files. Finally, use a CDN or other domains to host your resources and leverage domain sharding.

What is the purpose of cache busting and how can you achieve it?

Browsers have a cache to temporarily store files on websites so they don't need to be re-downloaded again when switching between pages or reloading the same page. The server is set up to send headers that tell the browser to store the file for a given amount of time. This greatly increases website speed and preserves bandwidth.

However, it can cause problems when the website has been changed by developers because the user's cache still references old files. This can either leave them with old functionality or break a website if the cached CSS and JavaScript files are referencing elements that no longer exist, have moved or have been renamed.

Cache busting is the process of forcing the browser to download the new files. This is done by naming the file something different to the old file.

A common technique to force the browser to re-download the file is to append a query string to the end of the file.

```
src="js/script.js" => src="js/script.js?v=2"
```

The browser considers it a different file but prevents the need to change the file name.

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

Name 3 ways to decrease page load (perceived or actual load time).

1. LocalStorage 2) Caching resources 3) dns-prefetch (sample below) 4) Keep resources on a CDN

If you jumped on a project and they used tabs and you used spaces, what would you do?

What is Flash of Unstyled Content? How do you avoid FOUC?

What is Flash of Unstyled Text? How do you avoid FOUT?

Explain what ARIA and screenreaders are, and how to make a website accessible.

Screen readers are software programs that provide assistive technologies that allow people with disabilities (such as no sight, sound or mouse-ing ability) to use web applications. You can make your sites more accessible by following ARIA standards such as semantic HTML, alt attributes and using [role=button] in the expected ways

What is the purpose of the `alt` attribute on images?

The `alt` attribute provides alternative information for an image if a user cannot view it. The `alt` attribute should be used to describe any images except those which only serve a decorative purposes, in which case it should be left empty.

Explain some of the pros and cons for CSS animations versus JavaScript animations.

Regarding optimization and responsiveness the debate bounces back and forth but, the concept is:

- CSS animations allows the browser to choose where the animation processing is done, CPU or the GPU. (Central or Graphics Processing Unit)
- That said, adding many layers to a document will eventually have a performance hit.
- JS animation means more code for the user to download and for the developer to maintain.
- Applying multiple animation types on an element is harder with CSS since all transforming power is in one property transform
- CSS animations being declarative are not programmable therefore limited in capability.

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

Cross Origin Resource Sharing. To address the fact that browsers restrict cross-origin HTTP requests initiated from within scripts. CORS gives web servers cross-domain access controls, which enable secure cross-domain data transfers.

Describe how you would create a simple slideshow page.

What's your favorite feature of Internet Explorer?

What UI, Security, Performance, SEO, Maintainability or Technology considerations do you make while building a web application or site?

Some people write good HTML, some people write bad HTML. What do you consider to be good HTML?

Tell me about a responsive project you have worked on recently?

Do you have any personal projects? Tell me about them.

Ways to decrease page load?

Ways to improve website performance

- Minimize HTTP Requests

    - Sites are mainly slow because of too many (or too large) HTTP requests. We can eliminate unnecessary request;
        - combined files: js to a file, css to a file
        - CSS sprites: CSS Sprites are the preferred method for reducing the number of image requests. Combine your background images into a single image and use the CSS background-image and background-position properties to display the desired image segment.

- Use a Content Delivery Network CDN

    - A CDN is essentially many optimized servers around the world that deliver web content to users based on their geographic location. This means big performance improvements for site users. Because, say, if a person accessing your site in India, they will be retrieving web content from a server nearby

- Optimize Images:

    - image sizes make a huge difference to site speed. The larger content/images, the slower the site. we could:
        - Changing the resolution: reducing the “quality” of the image (and thereby the file size)
        - Compressing the picture: increasing the efficiency of image data storage
        - Cropping the picture: when cropping, you are cutting out unneeded areas and thus making the image smaller in size

- Put Scripts at the Bottom:

    - Javascript files can load after the rest of your page. The simplest solution is to place your external Javascript files at the bottom of your page, just before the close of your body tag. Now more of your site can load before your scripts. Another method that allows even more control is to use the defer or async attributes when placing external .js files on your site.

        - Async tags load the scripts while the rest of the page loads, but this means scripts can be loaded out of order. Basically, lighter files load first. This might be fine for some scripts, but can be disastrous for others.

        - The defer attribute loads your scripts after your content has finished loading. It also runs the scripts in order. Just make sure your scripts run so late without breaking your site.

- Add an Expires or a Cache-Control Header

    - Web page designs are getting richer and richer, which means more scripts, stylesheets, images, and Flash in the page. A first-time visitor to your page may have to make several HTTP requests, but by using the Expires header you make those components cacheable. This avoids unnecessary HTTP requests on subsequent page views. Expires headers are most often used with images, but they should be used on all components including scripts, stylesheets, and Flash components.

- Gzip Components

    - Compression reduces response times by reducing the size of the HTTP response. Gzipping generally reduces the response size by about 70%.

- Put Stylesheets at the Top:

    - This is because putting stylesheets in the HEAD allows the page to render progressively.

- Avoid CSS Expressions

- Use GET for AJAX Requests:

    - Ajax is that it provides instantaneous feedback to the user because it requests information asynchronously from the backend web server

- Make JavaScript and CSS External:

    - Using external files in the real world generally produces faster pages because the JavaScript and CSS files are cached by the browser. JavaScript and CSS that are inlined in HTML documents get downloaded every time the HTML document is requested. This reduces the number of HTTP requests that are needed, but increases the size of the HTML document. On the other hand, if the JavaScript and CSS are in external files cached by the browser, the size of the HTML document is reduced without increasing the number of HTTP requests.

- Use get to ajax request:

- POST is implemented in the browsers as a two-step process: sending the headers first, then sending data. So it's best to use GET, which only takes one TCP packet to send (unless you have a lot of cookies).

- No 404s:
  - HTTP requests are expensive so making an HTTP request and getting a useless response (i.e. 404 Not Found) is totally unnecessary and will slow down the user experience without any benefit.
- Reduce Cookie Size:
  - HTTP cookies are used for a variety of reasons such as authentication and personalization. Information about cookies is exchanged in the HTTP headers between web servers and browsers. It's important to keep the size of cookies as low as possible to minimize the impact on the user's response time.
- Reduce DNS Lookups
- Minify JavaScript and CSS
- Avoid Redirects
- Remove Duplicate Scripts
- Configure Etags
- Make Ajax Cacheable
- Post-load Components
- Preload Components
- Reduce the Number of DOM Elements
- Minimize the Number of iframes
- Minimize DOM Access
- Optimize CSS Sprites
- Don't Scale Images in HTML
- Make favicon.ico Small and Cacheable
- Avoid Empty Image src

What tools do you use to test your code's performance?

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

How would you optimize a website's assets/resources?

Can you talk about how you have considered performance when developing a website?

What performance issues do you anticipate when building a website?

How would you go about troubleshooting a slow website?

What are some ways to prevent web browser caching?

How do you do browser compatibility testing?

What is your favorite browser? What sort of tools do you use to debug websites?

What do you think of “hacks”? When should they be used in your code and when should they be avoided?

What are the advantages of client side rendering vs. server side rendering? If you were building our site which would you use and why?

How do you test the performance of your code and/or web pages?

What is Flash of unstyled content?

How to avoid Flash of unstyled content.

How does the browser rendering engine work?

In order to render content the browser has to go through a series of steps:

- Document Object Model(DOM)
- CSS object model(CSSOM)
- Render Tree
- Layout
- Paint.

Can you explain the difference between GET and POST?

- https://stackoverflow.com/questions/3477333/what-is-the-difference-between-post-and-get

What is the DOM? How does the DOM work? Explain in as much detail as possible.

The DOM (Document Object Model) is a cross-platform API that treats HTML and XML documents as a tree structure consisting of nodes. These nodes (such as elements and text nodes) are objects that can be programmatically manipulated and any visible changes made to them are reflected live in the document. In a browser, this API is available to JavaScript where DOM nodes can be manipulated to change their styles, contents, placement in the document, or interacted with through event listeners.

- The DOM was designed to be independent of any particular programming language, making the structural representation of the document available from a single, consistent API.
- The DOM is constructed progressively in the browser as a page loads, which is why scripts are often placed at the bottom of a page, in the with a defer attribute, or inside a DOMContentLoaded event listener. Scripts that manipulate DOM nodes should be run after the DOM has been constructed to avoid errors.
- document.getElementById() and document.querySelector() are common functions for selecting DOM nodes.
- Setting the innerHTML property to a new value runs the string through the HTML parser, offering an easy way to append dynamic HTML content to a node.

What is the SAME ORIGIN POLICY

Comparision of browsers like Chrome, Firefox, Internet explorer, Safari etc

- Chrome:
    - Layout rendering `engine` Webkit.
    - JavaScript `engine` V8
- Firefox:
    - Layout rendering `engine` Gecko.
    - JavaScript `engine` Spider monkey
- Internet explorer:
    - Layout rendering engine``` Trident.
    - JavaScript `engine` Chakra
- Safari:
    - Layout rendering `engine` Webkit.
    - JavaScript `engine` JavascriptCore i.e Nitro

Open up your last project, explain your code and thought process.

```
// This is probably one of my favorite and most important questions.
// When I hire I like to hire candidates who are very passionate about what they do,
// and so programming isn’t something they just do at their job.
// They are actually passionate about it and pursue their own projects outside of work.
```

Teach me about something for the next 10 minutes.

```
// You are looking for them to select a topic they know
// I generally don’t care if they are technical or not
// and how well they communicate and break things down.
```

Explain the concept of cloud computing to my older (not-very-technical) mother.

```
// I actually had to do this in real life and it was harder than I expected.
// Another example is explain a database to someone’s grandparents.
```

In as much detail as you like, explain what happens when I type "google.com" into my browser's address bar and press enter.

How would you explain the Internet to a child?

Explain the importance of standards and standards bodies.

What happens when somebody hits the URL in browser?

What are HTTP methods? List all HTTP methods that you know, and explain them.

Why has it been better to serve site assets from multiple domains?

Do your best to describe the process from the time you type in a website's URL to it finishing loading on your screen.

Version Control

What is your preferred version control software?

Do you have experience working on a project versioned with Git?

What is your opinion on pull requests (and code reviews)?

What is the difference between Git and Github?

What would you consider an effective Git workflow?

What are build tools good for? Which ones are you familiar with?

Agile

Do you have experience working in an agile team?

Do you work in sprints, or using the kanban approach?

Do you have a preference, and why?

Work in Agile / Scrum?

Have you worked with QA? How you QA/test your code?

How did you communicate progress in your previous role? Did that process always work? What could have been done differently to keep everyone on the same page?

Tell me about a time when you had a miscommunication at work. What happened? If you could do it all over again would you alter your actions? Why or why not?

Have you ever disagreed with your boss or manager? What did you do? If you haven’t had this happen, imagine that it did, how would you handle this situation?

Give me an example of a time when you were able to communicate and work with another person even when they may not have personally liked you (or vice versa).

How well do you work on teams?

What did you like/not like about your last job?

What would your peers and instructors say about you?

Do you still write code in your job? Do you love it?

```
// This is more targeted at managers or leaders,
// since many do not need to be writing code to do their jobs.
// Hopefully this question could lead to an interesting conversation on role, etc.
```

Why do you want to work at [company name]? Have you used our products? Is there a particular area or feature that got you excited?

Give an example of when you completed a task without being asked. Can you give me another example? Another?

If you were hiring someone in this role, what would you look for? What sort of interview questions would you ask? Do you have a favorite question?

gohilumesh commented on Jan 26, 2017

Hi Mark, Would you mind adding this to your list : https://github.com/gohilumesh/front-end-interview-question

marko-jankovic commented on Mar 13, 2017 Author Owner

Hey **@gohilumesh** I'm trying to collect only HTML/CSS questions :) Thank you!

sarathkumar6 commented on Aug 7, 2018

@marko-jankovic Nice work. Very useful... **Thanks**.

s2c130050131527 commented on Oct 8, 2019

This list of questions is very helpful. I am amazed at the wide variety of concepts you have covered. Thank You.