CSS Questions

* **What is CSS selector specificity & how does it work?** 
  + The browser determines what styles to show on an element depending on the specificity of CSS rules.
  + There are four categories which define the specificity level of a selector

1. Inline styles, attached directly to the element to be styled

2. IDs,  unique identifier for the page elements

3. Classes, attributes and pseudo-classes.

4. Elements and pseudo-elements

* **What the difference between “resetting” & “normalizing” CSS? Which would you choose & why?** 
  + Resetting, strip all default browser styling on elements
  + Normalizing, preserves useful default styling rather then removing everything
* **Describe floats and how they work?** 
  + Float is a CSS positioning property
  + Floated elements remain a part of flow of the page and affect the positioning of other elements
  + Has an annoying collapsing effect on the parent container. Fixed by clearing the float in the container
* **how would you approach fixing browser-specific styling issues?** 
  + Use a separate style sheet that only loads when the specific browser is being used but this technique requires server-side rendering
  + Use libraries like Bootstrap that already handles these issues
  + Use Reset CSS or Normalize.css
* **Describe z-index & how stacking context is formed?** 
  + Z-index property controls the vertical stacking order of elements that overlap, it only affects elements that have a position value which is not static
  + Groups of elements with a common parent that move forward or backward together in the stacking order make up what is known as a stacking context.
  + New stacking contexts can be formed on an element in one of three ways

1. When a element is the root element of a document the <html> element

2. When an element has a position value other than static & z-index value there than auto

3. When a element has an opacity value less the 1

* **Describe Block Formatting Context (BFC) & how it works?** 
  + Block Formatting Context (BFC) is part of the visual CSS rendering of a web page in which block boxes are laid out.
  + Floats, absolute positioned elements, inline-block, table-cells, table-captions, and elements overflow other than visible establish new block formatting context
  + BFC is an HTML box that satisfies at least one of the following conditions

1. value of float is not none

2. Value of position is nether static nor relative

3. Value of display is table-cell, table-caption, inline-block, flex, or inline-flex

4. The value of overflow is not visible

* **Have you used or implemented media queries or mobile-specific layouts/CSS?** 
  + Yes for example turning a hamburger navigation into a fixed-bottom tab navigation after a certain breakpoint
* **What are the various clearing techniques & which is appropriate for what context?** 
  + Empty div method, <div style=“clear:both;”></div>
  + Clearfix method, creating additional class to clear floats
  + Overflow:auto or overflow:hidden, parents will establish a new block formatting context & expand to contains its floated children
  + In large project, I would write a utility .clearfix class and use them in places where I need it
* **Explain CSS sprites & how you would implement them on a page or site?** 
  + CSS sprites combine multiple images into one single larger image. It is a commonly used for icons
  + How to implement them?

1. Use a sprite generator that packs multiple images into one & generate the appropriate CSS for it.

2. Each image would have corresponding CSS class with background-image, background-position & background-size

* + Sprites reduce number of HTTP requests, advance downloading of assets that won’t be downloaded until needed
* **How do you serve your pages for feature-constrained browser? What techniques/processes do you use?** 
  + Graceful degradation, practice of building an application for modern browsers while ensuring it rains functional in older browsers
  + Progressive enhancement, practice of building an application for a base level of user experience but adding functional enhancements when a browser support it
  + I would use CSS feature queries support
  + Autoprefix for automatic vendor prefix insertion
* **What are the different ways to visually hide & make it available only for screen reader?** 
  + Visibility:hidden, but this is still in the flow of the page so it takes up space
  + Width:0; height:0, makes element not take up any space on the screen & not show it
  + Position:absolute; left: —99999px, position it outside of the screen
  + Text-indent: -99999px, but this only works on text within block elements
* **Have you ever used grid system if so what do prefer?** 
  + I have used flex, css grid and float based grid system.
  + I like using grid css because I feel like it has a lot of capabilities and since its the most latest and it available on almost all browser now
* **Are you familiar with styling SVG?** 
  + Yes there are several ways to color shapes like using inline CSS, an embedded CSS section, or an external CSS file
  + Basic coloring can be done by setting two attributes on the node fill & stroke
  + Fill sets the color inside the object & stroke sets the color of the line drawn around the object
* **Can you give an example of an @media property other than screen?** 
  + Yes there are four types of @media properties

1. All, for all media type devices

2. Print, for printers

3. Speech, for screenreader that reads the page out load

4. Screen, for computer screens, tablets, smart-phones etc

* **what are some of the “gotchas” for writing efficient CSS?** 
  + Understanding that the browser matches selectors from rightmost key selectors to left Browsers filter out elements in the DOM according to the key selector & travel up its parents elements to determine matches the shorter the length of the selector chain the faster browser can determine if that element matches the selector
* **what are the advantages/disadvantages of using CSS preprocessors?** 
  + Some advantages are

1. CSS is made more maintainable

2. Easy to write nested selectors

3. Variables for consistent theming, sharing theme files across different projects

4. Mixins to generate repeated CSS

5. Sass features like loops, list and maps can make configuration easier & less verbose

6. Splitting your code into multiple, CSS can do it but doing so will require a HTTP request to download each CSS file

* + Some disadvantages are

1. Requires tools for preprocessing & recompiling can be slow

* **Describe what you like & dislike about CSS preprocessor you have used?** 
  + Thing I like are

1. Variable, mixins, loops, list maps, coding splitting

2. Since less is written in javascript it works well with node

* + Thing I dislike are

1. When I use node-sass I have to frequently recompile when switching between node versions

* **How would you implement a web design comp that uses non-standard fonts?** 
  + Use @font-face and define font-family for different font-weights
* **Explain how browser determines what elements match a CSS selector?** 
  + Browsers match selectors from rightmost key selector to left.
  + Browsers filter out element in the DOM according to the key selector and travels up to its parent element to determine the match
  + For example with this selector p span, browsers firstly find all the <span> elements & travels up its parent all the way up to root to find <p> element
  + As soon as it find the <p> it know that <span> matches & can stop its matching
* **Describe pseudo-elements & discuss what they are used for?** 
  + Pseudo-element is a keyword added to a selector that lets you style a specific part of the selected elements
  + They can be used to decorate or adding elements to the markup without having to modify the mark up
  + :first-line & :first-letter can be used to decorate text
  + Its also used for clearfix class which add a zero-space element with clear:both
* **What is the difference between the “nth-of-type()” & “nth-child()” selectors?**
  + The nth-child() pseudo-class() is used to match an element based on a number which represents the elements position amongst its siblings
  + The nth-of-type() like nth-child() is used to match an element based on a number but this number represent the elements position within only those of its siblings that are of the same element type
* **Explain your understanding of the box model & how you would tell the browser in CSS to render your layout in different box model?** 
  + CSS box model describes the rectangular boxes that are created for elements in the document tree and laid out according to the visual formatting model. Each box has a content area (e.g. text, image, etc.) & optional surrounding padding, border & margin areas.
  + The CSS box model is responsible for calculating

1. How much space a block element takes up

2. Whether or not a border and margin overlaps or collapse

3. A box’s dimensions

* + The box model has the following rules

1. The dimensions of a block element are calculated by width, height, padding, borders & margins

2. If no height is specified a block element will be as high as the content it contains plus padding

3. If no width is specified a non-floated black element will expand to fit the width of its parent minus padding

4. The height of element is calculated by the contents height

5. The with of an aliment is calculated by the contents width

6. By default, paddings & borders are not part of the width & height of an element

* **What does \*{ box-sizing: border-box; } do? What are its advantages?**
  + By default element have box-sizing: content-box applied & only the content size is being accounted for
  + Box-sizing: border-box changes how the width & height of element are being calculated plus border & padding are also being included in the calculation
* **What is the CSS display property & can you give a few example of its use?**
  + The display property specifies the display behavior of an element
  + none, block, inline, inline-block, table, table-row, table-cell, list-item
* **What is the difference between relative, fixed, absolute & statically positioned element?**
  + A positioned element is an element whose computed position property is either relative, absolute, fixed or sticky
  + Static, default position; the element will flow into the page as it normally would. The top, right, bottom, left & z-index properties do not apply
  + Relative, elements position is adjusted relative to itself without changing which leaves a gap for the element where it would have been had it not been positioned
  + Fixed, element is removed from the flow of the page & positioned at a specific position relative to the viewport & doesn’t move when scrolled
  + Sticky, element is treated as relative position until it crosses a specified threshold then it get treated as fixed position
* **Can you explain the difference between coding a website to be responsive versus using a mobile-first strategy?**
  + These two approaches are not exclusive, making a website responsive means the same elements will respond by adapting its size or other functionality according to the devices screen size
  + Mobile-first strategy is a also responsive however it agrees we should default & define all the styles for mobile devices then specify for other devices later
* **How is responsive design different from adaptive design?**
  + Both responsive & adaptive design attempt to optimize the user experience across different devices, adjusting for different viewport sizes, resolutions, usage contexts, control mechanisms & so on
  + Responsive design work on the principle of flexibility, a single fluid website that can look good on any device. Responsive website use media queries, flexible grids and responsive images to create a user experience that flexes & changes based on a multitude of factors
  + 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 & layout based on a predefined set of viewport sizes and other characteristics. The site detects the type of device used & delivers the pre-set layout for the device.
* **Is there any reason you’d want 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 changes absolute positioning properties uses the CPU. Hence translate() is more efficient & will result in shorter paint times for smoother animations