CSS INTERVIEW QUESTIONS WITH ANSWERS

BASIC :

1. **What is CSS and why is it used?**

CSS (Cascading Style Sheets) is a stylesheet language used to describe the presentation of a document written in HTML or XML. It is used to control the layout, colors, fonts, and overall look of a web page.

2. **What are the different ways to apply CSS to a web page?**

Inline CSS: Directly in the HTML element using the style attribute.

Internal CSS: Within a <style> tag in the <head> section of an HTML document.

External CSS: Linked via an external stylesheet using the <link> tag.

Imported CSS: Using @import to import an external stylesheet within another CSS file.

3. **What is the difference between class and id selectors in CSS?**

class: Can be used multiple times on a page and is selected with a period (.) in CSS. Example: .className.

id: Must be unique within a page and is selected with a hash (#) in CSS. Example: #idName.

4. **How do you center a div horizontally?**

Using margin: 0 auto; on a block-level element with a specified width.
Example:

```
div {
   width: 50%;
   margin: 0 auto;
}
```

5. **What is the box model in CSS?**

The CSS box model describes the rectangular boxes generated for elements in the document tree and includes the content, padding, border, and margin.

6. **What are pseudo-classes in CSS?**

Pseudo-classes are keywords added to selectors that specify a special state of the selected elements. Examples include :hover, :focus, and :nth-child().

7. **Explain the difference between margin and padding.**

margin: The space outside an element's border, creating space between elements.

padding: The space inside an element's border, creating space between the content and the border.

8. **What is a CSS reset?**

A CSS reset is a set of rules that reset the default styling of HTML elements to ensure consistency across different browsers.

Example: normalize.css.

9. **How do you apply a style to multiple elements in CSS?**

By using a comma to separate multiple selectors.

Example:

```
h1, h2, p {
   color: blue;
}
```

10. **What are the various CSS units of measurement?**

Absolute units: px, cm, mm, in, pt, pc.

Relative units: em, rem, %, vw, vh, vmin, vmax.

INTERMEDIATE :

11. **Explain the concept of specificity in CSS.**

Specificity determines which CSS rule is applied when there is a conflict. It is calculated based on the number of ID selectors, class selectors, and type selectors in a rule. Inline styles have the highest specificity, followed by IDs, classes/attributes/pseudo-classes, and then element selectors.

12. **How does inheritance work in CSS?**

Inheritance allows child elements to inherit certain properties from their parent elements. Not all CSS properties are inherited; for example, text-related properties like color and font-family are inherited, while layout properties like margin and padding are not.

13. **What is the difference between em, rem, and px units?**

em: Relative to the font-size of the parent element.

rem: Relative to the font-size of the root element (html).

px: An absolute unit that represents one pixel on the screen.

14. **How do you implement a responsive design using CSS?**

By using media queries, flexible grid layouts, responsive images, and relative units like em, rem, vw, and vh.

15. **What is the difference between inline, block, and inline-block elements?**

inline: Elements flow within the content and do not start on a new line (e.g., <span>).

block: Elements start on a new line and take up the full width available (e.g., <div>).

inline-block: Elements flow within the content like inline elements but can have width and height set.

16. **What are media queries, and how do they work?**

Media queries are used to apply different styles for different devices or screen sizes. They work by using the @media rule to define conditions such as screen width, height, or device orientation.

17. **Explain the purpose of the z-index property.**

The z-index property controls the stacking order of elements on the z-axis (depth). Elements with a higher z-index are displayed on top of those with a lower z-index.

18. **What are CSS transitions, and how do they differ from animations?**

CSS transitions allow changes in CSS properties to occur smoothly over a specified duration. Transitions require a triggering event (like hover). CSS animations, defined with @keyframes, allow for more complex sequences of animations that can loop and do not require a triggering event.

19. **How do you create a CSS grid layout?**

By using the display: grid; property on a container and defining rows and columns with grid-template-rows and grid-template-columns. Items are placed within the grid using properties like grid-column, grid-row, grid-area, etc.

20. **What is the difference between absolute, relative, fixed, and sticky positioning?**

- relative: Positioned relative to its normal position.
- absolute: Positioned relative to its nearest positioned ancestor or the initial containing block.
- fixed: Positioned relative to the viewport, stays fixed when scrolling.
- sticky: Toggles between relative and fixed depending on the scroll position.

ADVANCED :

21. **How does the CSS Flexbox layout work, and when would you use it?**

Flexbox is a layout model that allows items to align and distribute space within a container. It is useful for creating flexible and responsive layouts without using floats or positioning. Use Flexbox for one-dimensional layouts (either row or column).

22. **What are CSS preprocessors, and how do they enhance CSS?**

CSS preprocessors like Sass, Less, and Stylus extend CSS with features like variables, nested rules, mixins, and functions. They compile down to standard CSS, making it easier to manage and maintain complex stylesheets.

23. **Explain the concept of CSS specificity hierarchy and how it can be overridden.**

Specificity hierarchy determines which CSS rules are applied when multiple rules match an element. It can be overridden using !important, inline styles, or by writing more specific selectors.

24. **How do CSS variables (custom properties) work?**

CSS variables are defined with the -- prefix and can be used throughout the stylesheet. They allow for easier theme management and dynamic styling.

Example:

```
:root {
   --main-color: #3498db;
}
body {
   color: var(--main-color);
}
```

25. **What are CSS combinators, and how are they used?**

Combinators describe the relationship between two selectors. Types of combinators include:

Descendant ( ): div p selects all <p> elements inside a <div>.

Child (>): div > p selects all <p> elements that are direct children of a <div>.

Adjacent sibling (+): h1 + p selects the first <p> element that is immediately preceded by an <h1>.

General sibling (~): h1 ~ p selects all <p> elements preceded by an <h1>.

26. **Explain the purpose and use of @keyframes in CSS animations.**

The @keyframes rule defines the stages of an animation by specifying keyframes (points in the animation) and the styles that should be applied at each stage.

Example:

```
@keyframes fadeIn {
   from { opacity: 0; }
   to { opacity: 1; }
}
```

27. **What is critical CSS, and why is it important for performance?**

Critical CSS is the minimal amount of CSS required to render the above-the-fold content of a page. By inlining critical CSS in the <head>, you reduce render-blocking resources, improving page load time.

28. **How do you implement dark mode in a website using CSS?**

Using CSS variables for color themes and media queries for detecting the user's preferred color scheme:

```
:root {
   --background-color: white;
```

```
  --text-color: black;
}
@media (prefers-color-scheme: dark) {
  :root {
    --background-color: black;
    --text-color: white;
  }
}
body {
  background-color: var(--background-color);
  color: var(--text-color);
}
```

29. **Explain how to create a responsive image gallery using CSS Grid.**

Define a grid container with responsive columns using grid-template-columns and auto-fit or auto-fill combined with minmax.

Example:

```
.gallery {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(150px, 1fr));
  gap: 10px;
}
```

```
.gallery img {
    width: 100%;
    height: auto;
}
```

30. **What is the difference between CSS Modules and traditional CSS?**

CSS Modules are a way to modularize and scope CSS by generating unique class names at build time, preventing class name collisions. Traditional CSS does not provide automatic scoping, which can lead to global styles affecting unintended elements.