**Term Test 2 Study Notes**

**Four Critical Issues in Web Development**

**Usability:** Refers to the ease with which users can navigate and interact with a website. A usable website is intuitive, efficient, and provides a positive experience for the user.

**Cross-browser Compatibility:** Ensuring that a website works and looks consistent across different browsers. Differences in how browsers interpret code can lead to inconsistencies.

**User Accessibility:** Websites should be designed so that everyone, including people with disabilities, can use them. This includes screen readers, keyboard navigation, and alternative text for images.

**SEO (Search Engine Optimization) & RWD (Responsive Web Design):** SEO involves optimizing a site to rank better in search engines, while RWD ensures the site looks good on all screen sizes and devices.

**HTML Semantic Elements**

Semantic elements clearly describe their meaning in a way that both the browser and developers can understand. Examples include:

* <header> for defining the top section of a page.
* <nav> for navigation links.
* <article> for independent, self-contained content. These elements improve accessibility, SEO, and code readability.

**Absolute URL vs. Relative URL**

**Absolute URL:** Specifies the full path, including the protocol (e.g., https://example.com/page.html).

**Relative URL:** Provides the path relative to the current page (e.g., /about or ../page.html). Absolute URLs are useful for linking to external resources, while relative URLs are better for internal navigation.

**CSS Box Model**

The CSS box model describes how elements are sized and spaced. It consists of:

**Content:** The actual text or image.

**Padding:** Space between the content and the border.

**Border:** Surrounds the padding and content.

**Margin:** Space outside the border, separating it from other elements.

**External, Embedded, and Inline Style Sheets**

**External Style Sheets:** CSS is written in a separate file (.css), which keeps HTML clean and is easy to maintain across multiple pages.

**Embedded Style Sheets:** CSS is written within the <style> tags in the HTML <head>. This is useful for small, single-page websites.

**Inline Style Sheets:** CSS is applied directly to HTML elements using the style attribute. This should be avoided as it reduces maintainability.

**JavaScript Shiv**

A JavaScript shiv (or polyfill) is a script used to ensure that older browsers can understand HTML5 elements, such as <header> and <section>. To use it, include the script in your HTML:

<!--[if lt IE 9]>

<script src="https://html5shiv.googlecode.com/svn/trunk/html5.js"></script>

<![endif]-->

**Absolute vs. Relative Measurements in CSS**

CSS sizes can be specified with absolute units (e.g., px, cm) or relative units (e.g., em, %):

**Pseudo-Class Selectors in CSS**

Pseudo-class selectors target elements based on their state or position, such as :hover for mouse-over actions or :nth-child() for targeting specific child elements. Example:

a:hover { color: red; }

**Style Precedence in CSS**

When multiple styles apply to the same HTML element, browsers follow a process to determine which style takes precedence:

**Importance**: Styles marked with !important take precedence.

**Specificity**: The more specific a selector, the higher its priority.

**Source Order**: The one that appears later in the CSS or HTML is applied.

**Floating Images in CSS**

To float an image to the left and allow block elements to flow to the right:

img { float: left; margin-right: 10px; }

To stop the flow of floating elements, use the clear property:

.clearfix { clear: both; }

This stops content from wrapping around the floating element.

**CSS Box Model Properties**

**Border Style**: Specifies the style of the border.

**Border Width**: Controls the thickness of the border.

**Border Color**: Sets the color of the border. It is possible to define borders for each side of the box separately using properties like border-top, border-right, border-bottom, and border-left.

**Margin**: Defines the space *outside* the border.

**Padding**: Defines the space *inside* the border.

**Key CSS Concepts**

**Fixed Layout vs. Fluid Layout**: Fixed layout uses absolute pixel values, while fluid layout uses relative units to allow for a more responsive design.

**Shiv vs. Shim**: A shiv is used to add support for new HTML5 elements in older browsers, while a shim often refers to a piece of code that fills in missing functionality in older environments.

**ID Selector vs. Class Selector**: The ID selector (#id) is unique for a single element on a page, whereas the class selector (.class) can be applied to multiple elements.

**Relational Selectors**:

* **Descendant**: Selects elements that are descendants of a specified ancestor (e.g., div p).
* **Child**: Selects only direct children (e.g., div > p).

**Sibling Selectors**:

* **General** (~): Selects all siblings that follow a specified element.
* **Adjacent** (+): Selects the element that immediately follows a specified element.

**Absolute Positioning**: Positions the element relative to its nearest positioned ancestor.

**Relative Positioning**: Positions the element relative to its normal position.

**Fixed Positioning**: Keeps the element fixed relative to the viewport.

**Creating a Multi-Column Web Page in CSS**

<!DOCTYPE html>

<html lang="en">

<head>

<style>

.column { float: left; width: 33.33%; }

.row::after { content: ""; display: table; clear: both; }

</style>

</head>

<body>

<div class="row">

<div class="column">Column 1</div>

<div class="column">Column 2</div>

<div class="column">Column 3</div>

</div>

</body>

</html>

**Advantages of Using CSS for Layout**

**Separation of Concerns**: CSS allows for separation of style and structure, making maintenance easier.

**Flexibility**: CSS provides more flexibility and responsiveness for modern web design.

**Accessibility**: CSS-based layouts are more accessible, as they allow screen readers and search engines to better interpret page content.

**VS Code and Web Development Trends**

**VS Code**: Is a popular text editor for web development due to its extensions, integrated Git, and debugging tools.

**Web Development Trends**: Current trends include increasing emphasis on responsive design, use of CSS Grid and Flexbox for layouts, and growing use of frameworks like React and Vue.js.