KaozhengPro

IT認證考試題庫

專業平臺

考證寶提供最新考古題與模擬試題
協助您高效通過認證考試

www.kaozhengpro.com

Exam : **Web Development Applications**

Title : WGU Web Development Applications (KVO1)

Version : DEMO

1.What is the purpose of cascading style sheets (CSSs)?
A. Structuring and describing web page content
B. Providing functionality that was previously provided by plug-ins
C. Changing the content of a web page dynamically
D. Setting rules to define how page content looks when rendered
Answer: D
Explanation:
Cascading Style Sheets (CSS) are used to control the presentation of web pages, including aspects such as layout, colors, fonts, and other visual styles. They are a cornerstone technology of the World Wide Web, along with HTML and JavaScript.
Here’s a detailed breakdown:
Purpose of CSS: CSS is designed to enable the separation of document content (written in HTML or a similar markup language) from document presentation, including elements such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting, and reduce complexity and repetition in the structural content.
Setting Visual Rules: CSS allows developers to define rules that specify how different elements of a web page should be displayed. For example, CSS rules can change text color, font size, spacing between elements, and even the overall layout of the web page. These rules are applied by the browser to render the web page according to the defined styles.
Cascading Nature: The term "cascading" in CSS refers to the process of combining multiple style sheets and resolving conflicts between different CSS rules. This allows developers to use different sources of style information, which can be combined in a hierarchical manner. For instance, a browser style sheet, an external style sheet, and inline styles can all contribute to the final rendering of a web page.
Benefits of CSS:
Consistency: By using CSS, developers can ensure a consistent look and feel across multiple web pages.
Maintainability: CSS makes it easier to update the visual presentation of a web page without altering the HTML structure. This is particularly useful for maintaining large websites.
Reusability: CSS rules can be reused across multiple pages, reducing redundancy and making it easier to implement changes globally.
Examples of CSS:
css
Copy code
body {
background-color: lightblue;
}
h1 {
color: navy;
margin-left: 20px;
}
In this example, the body element is given a light blue background color, and the h1 element is styled with a navy color and a left margin of 20 pixels.
Reference: MDN Web Docs on CSS

W3C CSS Specifications

2.Which feature was introduced in HTML5?
A. Addition of CSS in the HTML file
B. Adherence to strict XML syntax rules
C. Ability to hyperlink to multiple web pages
D. Native drag-and-drop capability
Answer: D
Explanation:
HTML5 introduced several new features that enhanced web development capabilities significantly. One of the notable features is the native drag-and-drop capability.
Native Drag-and-Drop Capability:
Description: HTML5 allows developers to create drag-and-drop interfaces natively using the draggable attribute and the DragEvent interface. This means elements can be dragged and dropped within a web page without requiring external JavaScript libraries.
Implementation:
Making an Element Draggable: To make an element draggable, you set the draggable attribute to true:
Drag me!
Handling Drag Events: You use event listeners for drag events such as dragstart, dragover, and drop:
document.getElementById("drag1").addEventListener("dragstart", function(event) {
event.dataTransfer.setData("text", event.target.id);
});
document.getElementById("dropzone").addEventListener("dragover", function(event) {
event.preventDefault();
});
document.getElementById("dropzone").addEventListener("drop", function(event) {
event.preventDefault();
var data = event.dataTransfer.getData("text");
event.target.appendChild(document.getElementById(data));
});
Example: This example demonstrates a simple drag-and-drop operation:
html
Copy code
Drag me!
Drop here
Reference: W3C HTML5 Specification - Drag and Drop
MDN Web Docs - HTML Drag and Drop API
HTML5 Doctor - Drag and Drop
HTML5's native drag-and-drop feature streamlines the process of creating interactive web applications by eliminating the need for third-party libraries, thus making it a powerful addition to the HTML standard.

3.Which structure tag should a developer use to place contact information on a web page?
A. <Aside>
B. <Main>

C. <Nav>
D. <footer>
Answer: D
Explanation:
The <footer> tag is used to define a footer for a document or a section. A footer typically contains information about the author of the document, contact information, copyright details, and links to terms of use, privacy policy, etc. It is a semantic element in HTML5, which means it clearly describes its meaning to both the browser and the developer.
Purpose of <footer>: The <footer> element represents a footer for its nearest sectioning content or sectioning root element.
It typically contains information like:
Contact information
Copyright information
Links to related documents
Information about the author
Usage Example:
<footer>
Contact us at: contact@example.com
© 2024 Example Company
</footer>
In this example, the <footer> tag encloses contact information and copyright details.
Semantic Importance: Using semantic elements like <footer> enhances the accessibility of the document and provides better context for search engines and other user devices.
Reference: MDN Web Docs on <footer>
W3C HTML5 Specification on <footer>

4.Which HTML tag should a developer use to create a drop-down list?
A. <Option>
B. <Section >
C. <Output>
D. <Select>
Answer: D
Explanation:
The <select> tag is used in HTML to create a drop-down list. It is used in conjunction with the <option> tags to define the list items within the drop-down menu.
Purpose of <select>: The <select> element is used to create a control that provides a menu of options. The user can select one or more options from the list.
Structure of Drop-down List:
The <select> element encloses the <option> elements.
Each <option> element represents an individual item in the drop-down list.
Usage Example:
<label for="cars">Choose a car:</label>
<select id="cars" name="cars">
<option value="volvo">Volvo</option>

<option value="saab">Saab</option>
<option value="fiat">Fiat</option>
<option value="audi">Audi</option>
</select>
In this example, the <select> tag creates a drop-down list with four options: Volvo, Saab, Fiat, and Audi.
Attributes of <select>:
name: Specifies the name of the control, which is submitted with the form data.
id: Used to associate the <select> element with a label using the <label> tag's for attribute.
multiple: Allows multiple selections if set.
Reference: MDN Web Docs on <select>
W3C HTML Specification on Forms

5.What should a developer correct before revalidating after a code validator reports multiple errors in code?
A. Only CSS errors
B. The last reported error
C. Only JavaScript errors
D. The first reported error
Answer: D
Explanation:
When using a code validator to check your HTML, CSS, or JavaScript code, it's essential to address errors in a systematic manner. The correct approach is to correct the first reported error before revalidating. This method is recommended because often, the first error can cause a cascade of subsequent errors or warnings. By fixing the first error, you may automatically resolve other errors that were reported later.
Reasoning:
Cascading Errors: The first error in the code might lead to multiple subsequent errors. Fixing it can sometimes resolve those cascading errors, reducing the overall number of errors in the next validation.
Logical Flow: Addressing errors in the order they appear maintains a logical flow and makes debugging more manageable.
Steps to Follow:
Step 1: Run the code through the validator.
Step 2: Identify the first reported error.
Step 3: Correct the first error.
Step 4: Revalidate the code to check if the error count has reduced or if new errors appear.
Step 5: Repeat the process until all errors are resolved.
Reference: W3C Markup Validation Service
MDN Web Docs - Debugging HTML
W3C CSS Validation Service