

New (2024) WGU Web-Development-Applications Exam Dumps [Q29-Q49]



New (2024) WGU Web-Development-Applications Exam Dumps [Q29-Q49]

New (2024) WGU Web-Development-Applications Exam Dumps

Best Way To Study For WGU Web-Development-Applications Exam Brilliant Web-Development-Applications Exam Questions PDF

NEW QUESTION 29

A web designer creates the following HTML code:

```
<div class="header" id="welcome">Welcome</div>  
<div class="header" id="intro">This is an introduction to cascading style sheets.</div>
```

Which CSS selector applies only to the first line?

- * #Welcome
- * .Welcome
- * #Header
- * .header

To apply CSS only to the first line of a particular HTML element, the ID selector #Welcome should be used as per the given HTML structure.

* CSS ID Selector: The ID selector is used to style the element with a specific id.

* Usage Example:

```
#Welcome {  
  
color: red;  
  
}
```

In this example, the #Welcome selector will apply the red color style only to the element with id=Welcome.

References:

* MDN Web Docs on ID Selectors

* W3C CSS Specification on Selectors

NEW QUESTION 30

Which method allows the end user to enter text as an answer to a question as the page loads?

- * Write
- * Confirm
- * Prompt
- * alert

The prompt method in JavaScript displays a dialog box that prompts the user for input, allowing the end user to enter text as an answer to a question when the page loads.

* prompt Method: The prompt method displays a dialog box with an optional message prompting the user to input some text. It returns the text entered by the user or null if the user cancels the dialog.

* Usage Example:

```
let userInput = prompt('Please enter your name:', 'Harry Potter');  
  
console.log(userInput);
```

In this example, a prompt dialog asks the user to enter their name, and the entered text is logged to the console.

References:

* MDN Web Docs on prompt

* W3C HTML Specification on Dialogs

NEW QUESTION 31

A file named application, Appcache contains the following content:

```
CACHE MANIFEST  
  
CACHE:  
default.html  
stylesheet.css  
functions.js  
logo.jpg
```

Which attribute should a developer set to application. Appcache so the four files will be cached?

- * The src attribute of the <script> element
- * The srcset attribute of the <source> element
- * The manifest attribute of the <html> element
- * The href attribute of the <Link> element

To specify a cache manifest file for an HTML document, the manifest attribute must be set in the <html> element. This attribute tells the browser to cache the files listed in the manifest for offline use.

* Cache Manifest:

* Purpose: Specifies resources that should be cached by the browser.

* Attribute: manifest is used in the <html> tag to link the cache manifest file.

* Example:

* Given the manifest file application.appcache:

CACHE MANIFEST

CACHE:

default.html

stylesheet.css

functions.js

logo.jpg

* HTML with the manifest attribute:

```
<html manifest="application.appcache">
```

```
</html>
```

* References:

* MDN Web Docs – Offline Web applications

* W3C HTML5 Specification – manifest attribute

NEW QUESTION 32

Which attribute displays help text in an input field without specifying an actual value for the input?

- * Default
- * For
- * Placeholder

* name

The placeholder attribute in an <input> element displays help text in the input field without specifying an actual value for the input. This text disappears when the user starts typing.

* Placeholder Attribute: This attribute provides a hint to the user about what type of information is expected in the field.

* Usage Example:

```
<input type="text" placeholder="Enter your name">
```

The input field will show "Enter your name" as help text.

References:

* MDN Web Docs on placeholder

* W3C HTML Specification on Input Placeholder

NEW QUESTION 33

Given the following CSS:

```
@keyframes custom {  
  0% {left: 0px; top: 0px;}  
  50% {left: 50%; top: 50%;}  
  100% {left: 0px; top: 100%;}
```

What is being configured?

- * Rendering to the canvas
- * Processing in the browser
- * Processing on a server

The given CSS configures properties that control the processing of animations directly in the browser.

* CSS Animations: CSS animations are processed by the browser's rendering engine. The animations defined by CSS are handled client-side and do not require server-side processing.

* Key Properties:

NEW QUESTION 34

A developer needs to apply a red font color to the navigation, footer, and the first two of three paragraphs on a website.

The developer writes the following code:

```
<nav class="standard"> </nav>  
<p class="standard"> </p>  
<p class="standard"> </p>  
<p id="specific"> </p>  
<footer class="standard"> </footer>
```

Which CSS archives this developer's goal?

- * Content
- * Padding
- * Margin
- * border

To apply a red font color to the navigation, footer, and the first two of three paragraphs on a website, the correct CSS would use the content attribute to achieve the desired styling. However, the term `content` isn't correct in the given context. The appropriate answer involves directly specifying styles for these elements using CSS selectors.

Here's the correct CSS:

```
nav, footer, p.standard:nth-of-type(1), p.standard:nth-of-type(2) {  
  
color: red;  
  
}
```

Explanation:

- * CSS Selectors: The selectors `nav`, `footer`, `p.standard:nth-of-type(1)`, and `p.standard:nth-of-type(2)` are used to target the specific elements. The `nth-of-type` pseudo-class is used to select the first and second paragraphs.
- * Applying Styles: The `color: red;` style rule sets the text color to red for the specified elements.

References:

- * MDN Web Docs on CSS Selectors
- * W3C CSS Specification on Selectors

NEW QUESTION 35

A web page has the following CSS code:

```
#black {  
  color:black;  
}
```

Which selector should a developer use to invoke the CSS?

- * Attribute
- * Style
- * id
- * class

To invoke the CSS provided, an ID selector is used. In CSS, the ID selector is used to style the element with the specific id attribute.

- * CSS ID Selector: The syntax for the ID selector is `#id`, where `id` is the id attribute value of the HTML element.
- * Usage Example:

```
#black {  
  
color: black;  
  
}
```

This CSS rule will apply the color: black; style to the element with id=#8221;black#8221;.

* Example in HTML:

```
<div id=#8221;black#8221;>This text will be black.</div>
```

References:

- * MDN Web Docs on ID Selectors
- * W3C CSS Specification on Selectors

NEW QUESTION 36

Given the following CSS code:

```
#name {  
text-align: left;  
}
```

Which type of selector is used?

- * Group
- * Class
- * Element
- * ID

The given CSS code uses the #name selector, which is an ID selector. The ID selector is used to style an element with a specific id attribute.

* ID Selector: In CSS, the ID selector is used to style the element with the specific id. The syntax for the ID selector is #id, where id is the id attribute value of the HTML element.

* Usage Example:

```
#name {  
  
text-align: left;  
  
}
```

This CSS rule will apply the text-align: left; style to the element with id=#8221;name#8221;.

* ID Selector Characteristics:

* An ID must be unique within a document, meaning it can be used only once per page.

* ID selectors are more specific than class selectors and element selectors.

* Example in HTML:

```
<div id="name">This is a div with ID "name";</div>
```

References:

* MDN Web Docs on CSS Selectors

* W3C CSS Specification on Selectors

NEW QUESTION 37

Which CSS transformation method should a developer use to reposition an element horizontally on the 2-D plane?

* Skewx (angle)

* Scale (x,y)

* Translate(n)

* Scale(n)

The translateX(n) method in CSS is used to move an element horizontally on the 2-D plane by a specified distance. This transformation repositions the element along the X-axis.

* translateX(n) Method: The translateX(n) function moves an element horizontally by n units. Positive values move the element to the right, while negative values move it to the left.

* Usage Example:

```
element {  
  
transform: translateX(100px);  
  
}
```

In this example, the element is moved 100 pixels to the right.

* Properties:

* n: This represents the distance to move the element. It can be specified in various units such as pixels (px), percentages (%), ems (em), etc.

References:

* MDN Web Docs on transform

* W3C CSS Transforms Module Level 1

NEW QUESTION 38

Given the following CSS statement:

```
body {  
  color: blue;  
}
```

Which code segment changes the font color when the viewport is 800 pixels wide or wider?

```
* @media screen and (min-width: 800px) {  
  body {  
    color: black;  
  }  
}
```

```
* @media min-width: 800px {  
  body {  
    color: black;  
  }  
}
```

```
* @media screen and (max-width: 800px) {  
  body {  
    color: black;  
  }  
} * @media max-width: 800px {  
  body {  
    color: black;  
  }  
}
```

To change the font color when the viewport is 800 pixels wide or wider, a media query with min-width:

800px is used. This ensures that the styles inside the media query are applied only when the viewport width is at least 800 pixels.

* CSS Media Queries:

* Syntax for Media Query:

```
@media screen and (min-width: 800px) {
```

```
  body {
```

```
    color: black;
```

```
  }
```

```
}
```

* Explanation: The min-width: 800px condition ensures that the styles are applied when the viewport is 800 pixels or wider.

* Example Analysis:

* Option A:

```
@media screen and (min-width: 800px) {
```

```
  body {
```

```
    color: black;
```

```
  }
```


}

* Correct. This applies the color: black; style to the body when the viewport is 800 pixels or wider.

* Option B:

```
@media min-width: 800px {
```

```
body {
```

```
color: black;
```

```
}
```

```
}
```

* Incorrect. Missing screen and which is required for a proper media query targeting screens.

* Option C:

```
@media screen and (max-width: 800px) {
```

```
body {
```

```
color: black;
```

```
}
```

```
}
```

* Incorrect. This applies the style when the viewport is 800 pixels or narrower.

* Option D:

```
@media max-width: 800px {
```

```
body {
```

```
color: black;
```

```
}
```

```
}
```

* Incorrect. This applies the style when the viewport is 800 pixels or narrower.

* References:

* MDN Web Docs [#8211; Using media queries](#)

* W3Schools – CSS Media Queries

The correct use of media queries ensures that the specified styles are applied only under the desired conditions, providing a responsive design.

NEW QUESTION 39

Which markup ensures that the data entered are either a five-digit zip code or an empty string?

- * `<input pattern=/d(5)–>`
- * `<input type=–number–; value=–s–>`
- * `Input required min=–s–; max=–s–>`
- * `Input class =–(0-9) (5)–>`

The pattern attribute in the `<input>` element is used to define a regular expression that the input value must match for it to be valid. The pattern `d{5}` ensures that the data entered is either a five-digit zip code or an empty string (if the required attribute is not used).

* Pattern Explanation:

* `d{5}`: Matches exactly five digits.

* This ensures that only a five-digit number or an empty string (if not required) is valid.

* Usage Example:

`<input type=–text–; pattern=–d{5}–; placeholder=–Enter a 5-digit zip code–>` This ensures that the input matches a five-digit zip code.

References:

- * [MDN Web Docs on pattern](#)
- * [Regular Expressions Documentation](#)

NEW QUESTION 40

A web designer inserts an image into a web page.

Which CSS attribute should this designer use to ensure that there is one pixel of space between the image and the surrounding elements?

- * Content
- * Padding
- * Margin
- * border

To ensure that there is one pixel of space between the image and the surrounding elements, the margin property should be used.

* CSS Margin Property: The margin property is used to create space around elements, outside of any defined borders.

* Usage Example:

`img {`

```
margin: 1px;
```

```
}
```

In this example, the `margin: 1px;` rule sets a margin of 1 pixel around the image, creating the desired space.

References:

- * MDN Web Docs on Margin
- * W3C CSS Specification on Margin

NEW QUESTION 41

What is the default behavior of overlay elements?

- * Last element listed appears on top
- * Last element listed appears on bottom
- * First element listed is invisible
- * First element listed is transparent

In CSS, when elements overlap, the default behavior is that the last element listed in the HTML document appears on top.

* Stacking Context:

* Default Behavior: Elements are stacked in the order they appear in the HTML. The last element in the document tree is rendered on top.

* z-index: You can control stacking order using the z-index property, but without it, the default order applies.

* Example:

* Given HTML:

```
<div style="position: absolute; width: 100px; height: 100px; background-color: red;"></div>
```

```
<div style="position: absolute; width: 100px; height: 100px; background-color: blue;"></div>
```

* The blue div will be on top of the red div because it appears later in the HTML document.

* References:

- * MDN Web Docs – Stacking context
- * W3C CSS Positioned Layout Module Level 3

By understanding these fundamental CSS concepts, developers can create more effective and visually appealing web layouts.

NEW QUESTION 42

Which feature was introduced in HTML5?

- * Addition of CSS in the HTML file

- * Adherence to strict XML syntax rules
- * Ability to hyperlink to multiple web pages
- * Native drag-and-drop capability

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`:

```
<div id="drag1" draggable=true>Drag me!</div>
```

* 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

```
<div id="drag1" draggable=true>Drag me!</div>
```

```
<div id="dropzone" style="width: 200px; height: 200px; border: 1px solid black;>Drop here</div>
```

* References:

* 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.

NEW QUESTION 43

Given the following JavaScript code:

```
var x;  
x = 6;  
document.getElementById("demo").innerHTML = x;
```

What happens when this code is executed?

- * The object with the id demo is changed to x.
- * The object with the id innerHTML is changed to 6.
- * The object with the demo is changed to 6.
- * The object with the id innerHTML is changed to x.

The provided JavaScript code sets a variable x to 6 and then assigns this value to the innerHTML of the HTML element with the id demo.

```
var x;
```

```
x = 6;
```

```
document.getElementById("demo").innerHTML = x;
```

Explanation:

- * Variable Declaration: `var x;` declares the variable x.
- * Variable Assignment: `x = 6;` assigns the value 6 to the variable x.
- * DOM Manipulation: `document.getElementById("demo").innerHTML = x;` finds the element with id demo and sets its innerHTML to the value of x, which is 6.

Outcome:

- * The content of the HTML element with the id demo will be changed to 6.

References:

- * MDN Web Docs on `getElementById`
- * W3C JavaScript DOM Manipulation

NEW QUESTION 44

What is the purpose of cascading style sheets (CSSs)?

- * Structuring and describing web page content

- * Providing functionality that was previously provided by plug-ins
- * Changing the content of a web page dynamically
- * Setting rules to define how page content looks when rendered

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.

References:

- * MDN Web Docs on CSS
- * W3C CSS Specifications

NEW QUESTION 45

Which code segment correctly defines a function in JavaScript?

- * `Void addNumbers(a, b)`
- * `Function addNumbers (a, b)`
- * `Void addNumber(int a, int b)`
- * `Function addNumber (in a, int b)`

In JavaScript, functions are defined using the function keyword followed by the name of the function, a set of parentheses (), and a block of code enclosed in curly braces {}.

* Function Definition Syntax:

* Correct Syntax:

```
function addNumbers(a, b) {
```

```
// function body
```

```
}
```

* Explanation:

- * `function`: Keyword to define a function.
- * `addNumbers`: Name of the function.
- * `(a, b)`: Parameters for the function.
- * `{ ... }`: Function body containing the code to be executed.

* Incorrect Options:

- * A. `Void addNumbers(a, b)`: JavaScript does not use void to define functions.
- * C. `Void addNumber(int a, int b)`: JavaScript does not use void or type declarations (int).
- * D. `Function addNumber (in a, int b)`: JavaScript functions do not use type declarations.

* References:

* MDN Web Docs – Functions

* W3Schools – JavaScript Functions

NEW QUESTION 46

Which code segment contains a conditional expression?

* `var result = dataCount;`

* `var result = count++ > 10;`

* `var result = count > 10 ? "Done" : getResult();`

* `var result = dataCount++ * 1000`

* Example Analysis:

* Option A:

```
var result = dataCount;
```

* This is a simple assignment, not a conditional expression.

* Option B:

```
var result = count++ > 10;
```

* This is a comparison but not a complete conditional expression.

* Option C:

```
var result = dataCount++ * 1000 === 3000000;
```

* This is an arithmetic operation followed by a comparison but not a complete conditional expression.

* Option D:

```
javascript
```

Copy code

```
var result = count >= 10 ? &#8220;Done&#8221; : getResult();
```

* This is a conditional (ternary) expression. If count is greater than or equal to 10, result will be “Done”; otherwise, it will call getResult().

* References:

* MDN Web Docs – Conditional (ternary) operator

* W3Schools – JavaScript Conditions

NEW QUESTION 47

Given the following CSS code:

```
body {  
  background-color : white;  
}  
  
,name {  
  color : black  
}
```

Which portion is the rule?

```
* body {  
  background-color : white;  
}  
* background-color : white;  
* color : black  
* .name
```

In CSS, a rule is composed of a selector and a declaration block. The rule is the entire part that defines how styles should be applied to elements matching the selector.

* CSS Rule Components:

* Selector: Identifies the HTML elements to be styled (e.g., body, .name, #item).

* Declaration Block: Contains one or more declarations, each consisting of a property and a value, enclosed in curly braces {}.

* Example Analysis:

* Option A:

```
body {  
  
background-color: white;  
  
}
```

This is the full rule, consisting of the selector body and the declaration block { background-color: white; }.

* Option B:

```
background-color: white;
```

This is just a declaration, not a complete rule.

```
background-color: white,
```

This is an incorrect declaration due to the comma, and not a complete rule.

* Option D:

```
color: black
```

This is just a declaration, not a complete rule.

* References:

* W3C CSS Syntax and Basic Data Types Module Level 3

* MDN Web Docs – CSS Syntax

NEW QUESTION 48

What is the effect of using the in-line validation process during user input?

- * Increased computation on the server
- * Larger number of errors created
- * More successful form completion rates
- * Slower rate of form completion

Inline validation during user input provides immediate feedback to the user about the correctness of their input. This real-time feedback helps users correct mistakes as they go, leading to higher rates of successful form completion.

* **Inline Validation:** Inline validation refers to checking the validity of input data as it is entered, rather than waiting until the form is submitted. This helps users to receive immediate feedback and correct errors on the spot.

* **Benefits:**

- * **Immediate Feedback:** Users can correct mistakes immediately, which helps reduce frustration and confusion.
- * **Increased Success Rates:** By providing real-time feedback, users are more likely to complete the form correctly, resulting in higher completion rates.
- * **User Experience:** Improves the overall user experience by making the process more interactive and user-friendly.

References:

- * Nielsen Norman Group on Inline Validation
- * MDN Web Docs on Constraint Validation

NEW QUESTION 49

Given the following code:

```
var a = "8";  
var b = "6";  
var c = a + b;  
console.log(c);
```

What does the console display as output?

- * 8
- * 86
- * 14

Given the code segment:

```
var a = 8 * 8;
```

```
var b = 6 * 8;
```

```
var c = a + b;
```

```
console.log(c);
```


This code concatenates two strings.

* Explanation:

* `var a = '8';`: Variable a is assigned the string '8';

* `var b = '6';`: Variable b is assigned the string '6';

* `var c = a + b;`: The + operator concatenates the two strings, resulting in '86';

* `console.log(c);`: Outputs the value of c to the console.

* Output:

* The console displays '86'; because it concatenates the two string values.

* References:

* MDN Web Docs 'String'

* W3Schools 'JavaScript String'

Updated Verified Pass Web-Development-Applications Exam - Real Questions and Answers:

<https://www.examlabs.com/WGU/Courses-and-Certificates/best-Web-Development-Applications-exam-dumps.html>