

Web Technology Practice Questions

HTML5

1. Define the term 'HTML5'.
2. List the basic elements of HTML5.
3. List the form elements in HTML5.
4. Name the media elements in HTML5.
5. Identify the basic elements of HTML5 used to create a user interface.
6. Classify the basic elements of HTML5 used in creating a webpage.
7. Apply your knowledge of HTML5 form elements to create a simple form with a text field for name, email field, and a submit button.
8. Demonstrate how media elements in HTML5 can be used to embed a video in a webpage.
9. Prepare a static website using HTML5 and XHTML.
10. Assess the importance of HTML5 form elements in creating a user-friendly web application.
11. Estimate the level of user engagement that can be achieved through the use of media elements in HTML5.
12. Build a personal resume website using HTML5 and XHTML.
13. Apply your understanding of HTML5 basic elements to create a simple webpage layout. Include at least one header, paragraph, and image element.
14. Demonstrate the use of form elements in HTML5 by creating a simple contact form that includes fields for name, email, and message.
15. Create a webpage that uses HTML5 media elements to display a video and an audio clip.
16. Design a simple static website using HTML5 and XHTML.
17. Design a static 'Coming Soon' page using HTML5 and XHTML.
18. Assemble a static website using HTML5 and XHTML that includes form elements, media elements, and graphical representations using Canvas and SVG. Include CSS3 animations and transformations.

CSS3

1. Write the definition of 'CSS'.
2. Name the types of CSS.
3. List the properties of the CSS box model.
4. Illustrate how CSS3 animations can be applied to create dynamic webpages.
5. Investigate how CSS3 selector strings, box models, and text properties can be used to style a webpage.
6. How do CSS3 selectors, the box model, and text properties work together to control layout and presentation in modern web design?
7. Create a new CSS3 animation that changes the color and size of a text when hovered over.
8. Create a CSS3 box model with a content width and height of 100px, padding of 10px, border of 5px, and margin of 15px.
9. Evaluate the impact of using CSS3 selector strings, box models, and text properties in website design.
10. Create a CSS3 code snippet to style a paragraph element with id 'info' with font size 20px, font color blue and background color yellow.
11. Demonstrate how to use CSS to style a webpage. Include at least three different CSS properties.
12. Modify the following HTML code to include a CSS3 animation: `<div id='myDiv'>Hello, world!</div>`

13. Investigate the following CSS code: ``body { color: red; } h1 { color: blue; }`` What color will the text of an h1 element be?
14. Distinguish between relative and absolute positioning in CSS.
15. Analyze the following CSS code: ``#myDiv { color: red; } .myClass { color: blue; }`` If an element has both the id 'myDiv' and the class 'myClass', what color will its text be?
16. Assess the role of CSS3 selector strings in the styling of a webpage.
17. Develop a web page using CSS3 3D transformations and animations.

XML and XSLT

1. Label the parts of a basic XML structure.
2. Interpret the role of namespaces in XML.
3. Differentiate between XML and JSON data formats.
4. Validate the correctness of the following XML code snippet: ``<book><title>XML Developer's Guide</title><author>Gambardella, Matthew</author><price>44.95</price></book>``
5. Construct an XML document representing a book store, containing at least 3 books with their titles, authors, and prices.
6. Transform the following XML data using XSLT:
``<students><student><name>John</name><grade>A</grade></student><student><name>Jane</name><grade>B</grade></student></students>``
7. Create an XML document that includes DTD and namespaces. Then, use XSLT to transform the XML data.
8. Justify the use of XML namespaces in avoiding element name conflicts.

JavaScript

1. Describe the role of JavaScript operators.
2. State the purpose of JavaScript control statements.
3. Identify the primary data types in JavaScript.
4. Distinguish between the different data types in JavaScript.
5. Contrast the use of JavaScript operators and control statements.
6. How can JavaScript be used to enhance user interaction through real-time form validation?
7. Demonstrate the use of JavaScript operators to manipulate data.
8. Illustrate how JavaScript operators can be used to manipulate data by creating a simple calculator that can perform addition, subtraction, multiplication, and division.
9. Prepare a JavaScript function that validates an email field in a form. The function should return true if the email is in the correct format and false otherwise.
10. Evaluate how JavaScript operators help in validating user input and processing data dynamically in client-side applications.
11. Justify the use of JavaScript control statements in controlling the flow of a web application.
12. Predict the output of the following JavaScript function: ``function test() { var x = 10; var y = x++; return y; }``
13. Develop a JavaScript function that calculates the sum of an array of numbers.
14. Compute the area of a rectangle using JavaScript. Assume the length is 5 units and the breadth is 3 units.
15. Predict the output of the following JavaScript code: ``var x = 10; var y = '5'; var z = x + y;``
16. Use JavaScript to write a function that takes two numbers as arguments and returns their sum.
17. Compute the result of the following JavaScript operation: ``var x = 5; var y = 2; var z = x y;``
18. Write a JavaScript function that calculates the factorial of a given number.

19. Solve the following problem using JavaScript: Write a function that takes an array of numbers and returns the largest number.
20. Interpret the following JavaScript code: ``var x = 10; var y = x++;`` What are the final values of x and y?
21. Differentiate between the `'=='` and `'==='` operators in JavaScript.
22. Examine the following JavaScript code: ``var x = 10; var y = '10'; var z = (x == y);`` What is the value of z?
23. Contrast the use of the `'var'` and `'let'` keywords in JavaScript.
24. Identify the error in the following JavaScript code: ``function sum(a, b) { return a + b } var x = sum(10, '5');``
25. Experiment with the following JavaScript code: ``var x = 10; var y = x++;`` What are the final values of x and y?
26. Differentiate between the `'getElementById'` and `'querySelector'` methods in JavaScript.
27. Estimate the efficiency of using JavaScript control statements in managing the flow of a program.
28. Justify the use of JavaScript objects in encapsulating related data and functions.
29. Predict the output of the following JavaScript code snippet: ``var x = 10; function myFunc() { var x = 20; console.log(x); } myFunc(); console.log(x);``
30. Create a new JavaScript object representing a student with properties name, age, and course.
31. Generate a JavaScript function that performs a specific task.
32. Formulate a JavaScript control statement that manipulates data.
33. Create a JavaScript object that encapsulates related data and functions.
34. Develop a JavaScript program that utilizes data types, variables, operators, control statements, functions, objects, and built-in objects.
35. Formulate a JavaScript program that manipulates the DOM and handles events.
36. Generate a JavaScript program that uses form handling and validations. Also, manipulate the DOM and handle events.

AJAX

1. Define the term 'AJAX'.
2. Differentiate between traditional web requests and AJAX-based requests.
3. Describe the role of AJAX in creating dynamic web applications.
4. Contrast the role of AJAX in creating dynamic webpages with traditional synchronous requests.
5. Relate the usage of AJAX in creating dynamic web pages by implementing a simple AJAX call that fetches data from a server and updates a part of the webpage without refreshing.
6. Examine how JQuery AJAX can be used to handle asynchronous requests.
7. Validate the effectiveness of using AJAX in creating dynamic web applications.
8. Construct a simple AJAX client-server architecture diagram.
9. Investigate the role of the XMLHttpRequest object in AJAX. How does it enable communication between the client and server?
10. Defend the use of AJAX in creating dynamic web content. Provide examples.
11. Generate an AJAX code snippet to load the content of a file named 'info.txt' into a div element with id 'myDiv'.
12. Generate a web page that uses AJAX to handle asynchronous requests and JQuery for DOM traversing, event handling, animations, and effects.

13. Create an XML document with namespaces and use XSLT to transform the data. Also, use AJAX to handle asynchronous requests to the server.

jQuery

1. Define the term 'jQuery'.
2. Discuss the advantages of using jQuery in modern web applications.
3. Show how jQuery can be used to handle a button click event.
4. Demonstrate how to use jQuery selectors to manipulate HTML elements.
5. Verify the role of jQuery in simplifying client-side scripting of HTML.
6. Create a jQuery animation that fades out a div when a button is clicked.
7. Formulate a jQuery code snippet to animate a div element with id 'myDiv' to slowly hide when a button is clicked.
8. Verify the functionality of the following jQuery code snippet: ``$(document).ready(function(){
$('button').click(function(){ $('p').hide(); }); });``
9. Select the appropriate jQuery animation effect for a given user interaction scenario.
10. Create a jQuery animation to enhance user interaction.
11. Assemble a jQuery event handler for a specific user action.
12. Modify an HTML element using jQuery selectors and methods.
13. Develop a web page that uses jQuery to handle events, apply animations, and traverse the DOM.
14. Design a web page that demonstrates the use of jQuery for animations, user interaction (event handling), and AJAX requests.

Bootstrap

1. State the purpose of the Bootstrap framework.
2. Compare the Bootstrap grid system with traditional CSS layout methods.
3. Interpret the role of the Bootstrap Grid in responsive web design.
4. Apply the Bootstrap framework to design a responsive webpage.
5. Create a static website using HTML5, CSS3, and Bootstrap. The website should have a navigation bar, a carousel, and a form.
6. Critique the use of Bootstrap framework in responsive web design. Provide examples.
7. Construct a web page using the Bootstrap framework. The page should include a user-friendly form created with Bootstrap form elements.
8. Transform an existing website to be responsive using the Bootstrap framework. The website should include a grid system, form elements, and CSS3 animations.
9. Design an online blog using HTML5, CSS3, and Bootstrap. The blog should include form handling and validations using JavaScript.

Servlets and JSP

1. Summarize the lifecycle of a servlet.
2. Explain the difference between JSP and servlets.
3. Use the servlet API to read servlet parameters.
4. Discuss the concept of Session Management in Servlets.
5. Explain the steps involved in the creation and destruction of a Servlet by a web container.
6. Plan and create a todo list using servlets and JDBC.
7. Construct a servlet that handles a specific request.
8. Modify a servlet to implement session management and connect to a database using JDBC.

9. Design and implement a Java application that performs CRUD (Create, Read, Update, Delete) operations on a 'Student' database table using JDBC.
10. Create a web page using JSP that includes processing, declarations, directives, expressions, implicit objects, and session tracking.
11. Transform a static web page into a dynamic one using JSP.
12. Build a JSP-based dynamic web application that tracks user sessions, uses standard JSP components, and connects to a database through Spring framework integration.
13. Illustrate how to build a servlet that handles session management, reads request parameters, and connects to a MySQL database using JDBC.
14. Develop a web page using JSP that includes implicit objects and session tracking. Also, use Java frameworks like Spring.
15. Construct a servlet that reads parameters using the servlet API, implements session management, and connects to a database using JDBC.

Java Frameworks

1. What are Java frameworks? Name a few commonly used ones.
2. Identify the role of Java frameworks like Spring and Hibernate in web application development.
3. Defend the use of JSP over servlets in web application development.
4. Build a JSP-based dynamic web application that tracks user sessions, uses standard JSP components, and connects to a database through Spring framework integration.
5. Develop a web page using JSP that includes implicit objects and session tracking. Also, use Java frameworks like Spring.

Web Servers and Client-Server Communication

1. Identify the basic internet protocols in web technology.
2. List the key internet protocols that support web browsing and data transfer over the Internet.
3. Describe the process of client-server communication and the role played by a web server in managing that communication.
4. Explain the role of a web server in the client-server communication model.
5. What is Apache Tomcat? Describe its function as a servlet container.
6. Explain the role of the Apache-Tomcat Server in web development.