**Chat history**

50 question with answers php interview question

ChatGPT

Sure, here are 50 PHP interview questions along with their answers:

**Basic Questions**

1. **What is PHP?**
   * PHP is a server-side scripting language designed primarily for web development but also used as a general-purpose programming language.
2. **What does PHP stand for?**
   * PHP stands for "Hypertext Preprocessor" (a recursive acronym).
3. **What is the difference between echo and print in PHP?**
   * **echo** can output multiple strings, while **print** can only output one string and always returns 1.
4. **How do you declare a variable in PHP?**
   * Variables in PHP are declared with a dollar sign **$**, followed by the variable name. Example: **$variableName = 'value';**.
5. **What are the data types supported by PHP?**
   * PHP supports several data types: Integer, Float, String, Boolean, Array, Object, NULL, Resource.
6. **How do you create a constant in PHP?**
   * Use the **define()** function. Example: **define("CONSTANT\_NAME", "value");**.
7. **What are superglobals in PHP?**
   * Superglobals are built-in variables that are always accessible, regardless of scope. Examples include **$\_GET**, **$\_POST**, **$\_SESSION**, **$\_COOKIE**, **$\_FILES**, **$\_REQUEST**, **$\_SERVER**, **$\_ENV**.
8. **What is the difference between == and === in PHP?**
   * **==** checks for value equality, while **===** checks for value and type equality.
9. **How do you include a file in PHP?**
   * Use **include 'filename.php';** or **require 'filename.php';**.
10. **What is the difference between include and require?**
    * **include** will emit a warning if the file cannot be included, while **require** will emit a fatal error.

**Intermediate Questions**

1. **What is a session in PHP?**
   * A session is a way to store information (in variables) to be used across multiple pages.
2. **How do you start a session in PHP?**
   * Use **session\_start();**.
3. **How can you destroy a session in PHP?**
   * Use **session\_destroy();**.
4. **What is an array in PHP?**
   * An array is a data structure that stores one or more similar type of values in a single value.
5. **What are the types of arrays in PHP?**
   * Indexed arrays, Associative arrays, and Multidimensional arrays.
6. **How do you create an associative array in PHP?**
   * Example: **$arr = array("key1" => "value1", "key2" => "value2");**.
7. **What is the use of foreach loop in PHP?**
   * **foreach** is used to loop through arrays.
8. **How do you connect to a MySQL database using PHP?**
   * Use the **mysqli\_connect()** function.
9. **What is PDO in PHP?**
   * PDO (PHP Data Objects) is a database access layer providing a uniform method of access to multiple databases.
10. **What is the difference between mysql\_connect and mysqli\_connect?**
    * **mysql\_connect** is deprecated; **mysqli\_connect** is improved and more secure, with support for prepared statements.

**Advanced Questions**

1. **What are prepared statements in PHP?**
   * Prepared statements are used to execute the same statement repeatedly with high efficiency and to prevent SQL injection.
2. **What is the use of strip\_tags() function in PHP?**
   * **strip\_tags()** strips HTML and PHP tags from a string.
3. **How can you prevent SQL injection in PHP?**
   * Use prepared statements with bound parameters, or use real escape string functions like **mysqli\_real\_escape\_string()**.
4. **What is the difference between GET and POST methods in PHP?**
   * **GET** appends data to the URL and is visible, while **POST** sends data through the HTTP header and is not visible.
5. **How do you handle file uploads in PHP?**
   * Use the **$\_FILES** superglobal array.
6. **What is cURL in PHP?**
   * **cURL** is a library that allows you to connect and communicate with different types of servers using various protocols.
7. **What are traits in PHP?**
   * Traits are a mechanism for code reuse in single inheritance languages like PHP.
8. **What is the purpose of json\_encode() and json\_decode()?**
   * **json\_encode()** converts a PHP array to JSON format, while **json\_decode()** converts JSON to a PHP array or object.
9. **How can you redirect a page in PHP?**
   * Use the **header("Location: url");** function.
10. **What are namespaces in PHP?**
    * Namespaces are a way to encapsulate items to avoid name conflicts.

**Expert Questions**

1. **What is Composer in PHP?**
   * Composer is a dependency management tool for PHP, allowing you to manage libraries required in your projects.
2. **How do you declare an abstract class in PHP?**
   * Use the **abstract** keyword. Example: **abstract class ClassName {}**.
3. **What is an interface in PHP?**
   * An interface is a contract that classes must follow by implementing all methods defined in the interface.
4. **What is the difference between an abstract class and an interface?**
   * An abstract class can have implemented methods and properties, while an interface can only have method declarations.
5. **How do you handle errors in PHP?**
   * Use **try-catch** blocks and error handling functions like **set\_error\_handler()** and **error\_reporting()**.
6. **What is the purpose of final keyword in PHP?**
   * The **final** keyword prevents child classes from overriding a method or inheriting from a class.
7. **What is PSR in PHP?**
   * PSR stands for PHP Standard Recommendation, which is a set of standards and guidelines to ensure interoperability of PHP components and frameworks.
8. **How do you create a singleton pattern in PHP?**
   * Ensure a class has only one instance by creating a private constructor, a private static instance variable, and a public static method to get the instance.
9. **What are magic methods in PHP?**
   * Magic methods are special methods like **\_\_construct()**, **\_\_destruct()**, **\_\_get()**, **\_\_set()**, **\_\_call()**, **\_\_toString()**, etc., that PHP automatically calls in certain situations.
10. **How do you autoload classes in PHP?**
    * Use the **spl\_autoload\_register()** function.

**Miscellaneous Questions**

1. **What is the difference between unlink() and unset() in PHP?**
   * **unlink()** deletes a file, while **unset()** destroys a specified variable.
2. **What is the use of isset() and empty() functions in PHP?**
   * **isset()** checks if a variable is set and is not NULL, while **empty()** checks if a variable is empty.
3. **How do you create a cookie in PHP?**
   * Use the **setcookie()** function. Example: **setcookie("name", "value", time()+3600);**.
4. **How do you retrieve a cookie value in PHP?**
   * Use the **$\_COOKIE** superglobal array.
5. **What is htaccess file in PHP?**
   * **.htaccess** is a configuration file for Apache web server used to alter its configuration on a per-directory basis.
6. **What is the header() function in PHP?**
   * The **header()** function sends a raw HTTP header to the client.
7. **How do you handle JSON data in PHP?**
   * Use **json\_encode()** to encode data to JSON and **json\_decode()** to decode JSON data.
8. **What is the explode() function in PHP?**
   * The **explode()** function splits a string by a specified delimiter into an array.
9. **How do you check if a file exists in PHP?**
   * Use the **file\_exists()** function.
10. **What is the purpose of ob\_start() and ob\_end\_flush() in PHP?**
    * **ob\_start()** turns on output buffering, while **ob\_end\_flush()** sends the buffered output to the browser.

**Java script interview questions** Top of Form

**Question 1**

What will be the output of the below code:

const array = [10, 20, 30, 40];

const result = array.map((num) => num / 2).filter((num) => num >= 15);

console.log(result);

**Question 2**

Find the issue in the below code snippet.

let counter = 0;

for (var i = 1; i <= 10; i++) {

counter+= i;

}

console.log(counter);

console.log(i);

**Question 3**

Analyze the below code. Do you see any issue? If yes, what is that issue?

const object1 = {

a: 10,

b: 20,

c: function () {

console.log(this.a + this.b);

},

};

const func = object1.c;

func();

**Question 4**

Create a JavaScript function that calculates the tip for a given bill amount and tip percentage. Bill amount and tip percentage will be input parameters while output will be calculated tip value.

**Question 5**

What will be the output of below code snippet:

function greetHello(name) {

return `Hello, ${name}!`;

}

console.log(greetHello("Brian"));

**Question 6**

 Will the below code return any error? If yes, identify the error.

function fetchData(callback) {

fetch('https://api.example.com/data')

.then(response => response.json())

.then(data => callback(null, data))

.catch(error => callback(error));

}

fetchData(function (error, data) {

if (error) {

console.log('Error:', error);

} else {

console.log('Data:', data);

}

});

**Question 7**

Implement a simple shopping cart system with features to add items, remove items and calculate the total price. Use objects to represent items, including properties for the item name, price and quantity. Implement features to add items to the cart, remove items and calculate the total cost.

**Question 8**

Analyze the below code snippet and advise what will be the output:

const person = {

firstName: "Helen",

lastName: "Ryan",

getFullName: function () {

return this.firstName + " " + this.lastName;

},

};

console.log(person.getFullName());

**Question 9**

Find the issue with the below code snippet:

setTimeout(function () {

console.log("This will be executed after 3 seconds");

}, 3000);

clearTimeout();

**Question 10**

What issue exists in the below code:

const testArray = [1, 2, 3];

testArray = [4, 5, 6];

console.log(testArray);

**Interview Questions for Mid-level JavaScript Developers**

When designing interview questions for mid-level JavaScript developers, you should prepare challenges to test the understanding of advanced JavaScript concepts and problem-solving skills. Some areas that should be considered for evaluation include functional programming, asynchronous programming, promises, working with APIs and advanced JavaScript features. Find below 10 coding challenges which are suited to mid-level JavaScript developers:

**Question 1**

What is the issue in the below code:

const fetchData = async () => {

const response = await fetch("https://api.samplewebsite.com/data");

const data = await response.json();

console.log(data);

};

fetchData();

**Question 2**

What will be the output of the below code:

const promise1 = Promise.resolve("One");

const promise2 = new Promise((resolve) => setTimeout(() => resolve("Two"), 1000));

const promise3 = Promise.reject("Three");

Promise.allSettled([promise1, promise2, promise3]).then((results) => console.log(results));

**Question 3**

Develop a simple URL shortener service using JavaScript. Implement a function that takes a long URL as an input parameter and the output will be a shortened URL. Create a reverse function as well. The reverse function takes the shortened URL and returns the original long URL. You can use simple in-memory objects to store the mapping between long and short URLs.

**Question 4**

Implement an autocomplete feature for a search input field. Given an array of words, write a function that suggests words based on the current input. The output of the function will be an array of suggested words that start with the input characters, limiting the number of suggestions (e.g., a maximum of 7 suggestions).

**Question 5**

What is the issue in the below code:

const obj = {

name: "Conner",

age: 27,

greet: () => {

console.log(`Hey, my name is ${this.name}`);

},

};

obj.greet();

**Question 6**

What will be the output of below code snippet:

const object1 = {

prop1: "value1",

prop2: {

prop3: "value3",

},

};

const newObj = { ...obj };

newObj.prop2.prop3 = "newValue3";

console.log(object1.prop2.prop3);

**Question 7**

Will the below code return any error? If yes, what will be the error?

class Bird {

constructor(name) {

this.name = name;

}

speak() {

console.log(`${this.name} makes a noise.`);

}

}

class Crow extends Bird{

speak() {

super.speak();

console.log(`${this.name} sings.`);

}

}

const crow = new Crow("Tim");

crow.speak();

**Question 8**

Develop a function that throttles another function, allowing it to be called at most once every specified interval (e.g., 300ms). The throttling function will have two input parameters. One will be the function to be throttled and the second will be the interval in milliseconds. The throttled function should be called with the same arguments as the original function.

**Question 9**

What is wrong with the below code:

const arr = [1, 2, 3, 4, 5];

const sum = arr.reduce((total, num) => total + num);

console.log(sum / arr.length);

**Question 10**

Design a simple meeting scheduler that finds the first available time slot for a meeting between two people. Given two arrays of busy time intervals and a meeting duration, create a function that returns the earliest available time slot for the meeting when both people will be available. Each interval is represented as an array of two integers, where the first integer is the start time and the second integer is the end time.

**Interview Questions for Expert JavaScript Developers**

When preparing coding challenges for expert-level JavaScript engineers, you should test the advanced features of the language and performance optimization techniques. Some of the areas to evaluate include advanced JavaScript features, code architecture, design patterns, performance optimization and security. Below we have presented 10 coding challenges for expert JavaScript developers:

**Question 1**

Is there any security vulnerability in the below code? If yes, identify it:

const username = document.getElementById('username').value;

const password = document.getElementById('password').value;

fetch('https://api.examplewebsite.com/login', {

method: 'POST',

body: JSON.stringify({ username, password })

})

.then(response => response.json())

.then(data => console.log(data))

.catch(error => console.log(error));

**Question 2**

Identify the output of the below code.

const testArray = [1, 2, 3, 4, 5];

const res = testArray.reduce((acc, curr) => {

if (curr % 2 === 0) {

return acc + curr;

}

return acc;

}, 0);

console.log(res);

**Question 3**

What is the possible performance issue in the below code?

const arr = [];

for (let i = 0; i < 1000000; i++) {

arr.push(Math.floor(Math.random() \* 1000));

}

**Question 4**

Suggest the output of the below code:

const arr = [1, 2, 3];

const object1 = { x: 1, y: 2, z: 3 };

console.log([...arr, ... object1]);

**Question 5**

Design a social media platform that contains features like sign up, creating a profile and posting status updates. Users should be able to follow other users and view their posts on a newsfeed.

**Question 6**

What is wrong with the below call to the API?

fetch('https://api.example.com/data')

.then(response => {

if (!response.ok) {

throw new Error('Network response was not ok');

}

return response.json();

})

.then(data => console.log(data))

.catch(error => console.log(error));

**Question 7**

What will be the output of below code snippet?

const promise1 = Promise.resolve(One);

const promise2 = Promise.resolve(Two);

Promise.all([promise1, promise2]).then(([result1, result2]) => console.log(result1 + ' ' + result2));

**Question 8**

Design an online code editor where users can write, save and run JavaScript code. The editor should include features like syntax highlighting, auto-completion and error checking.

**Question 9**

The below code snippet uses closures to implement a counter. How will you optimize it to minimize memory usage:

function counter() {

let count = 0;

return function() {

count++;

console.log(count);

}

}

const increment = counter();

increment(); // 1

increment(); // 2

increment(); // 3

**Question 10**

Develop a fitness tracker application where users can enter their daily exercise routines and track their progress over time. The application should allow users to set goals, view their progress and receive reminders.

**Basic Questions**

1. **What is JavaScript?**
   * JavaScript is a high-level, dynamic, untyped, and interpreted programming language that is commonly used to create interactive effects within web browsers.
2. **What is the difference between let, const, and var?**
   * **var** is function-scoped, while **let** and **const** are block-scoped. **const** is used for variables that should not be reassigned, whereas **let** can be reassigned.
3. **What are data types in JavaScript?**
   * JavaScript supports the following data types: Undefined, Null, Boolean, Number, String, Symbol, Object.
4. **How do you declare a variable in JavaScript?**
   * Variables can be declared using **var**, **let**, or **const**. Example: **let x = 10;**.
5. **What is the difference between == and ===?**
   * **==** checks for value equality, performing type conversion if necessary. **===** checks for both value and type equality without type conversion.
6. **What are functions in JavaScript?**
   * Functions are blocks of code designed to perform a particular task and are executed when they are invoked (called).
7. **What is an arrow function?**
   * An arrow function is a shorter syntax for writing function expressions using the **=>** syntax. Example: **const add = (a, b) => a + b;**.
8. **What is a closure?**
   * A closure is a function that retains access to its lexical scope, even when the function is executed outside that scope.
9. **What is the this keyword?**
   * The **this** keyword refers to the object it belongs to. Its value depends on where it is used: in a method, it refers to the owner object; alone, it refers to the global object.
10. **What is an event in JavaScript?**
    * An event is an action or occurrence that happens in the browser, like a click or keypress, to which you can respond using event handlers.

**Intermediate Questions**

1. **What is the Document Object Model (DOM)?**
   * The DOM is a programming interface for web documents. It represents the page so that programs can change the document structure, style, and content.
2. **What is event delegation?**
   * Event delegation is a technique of handling events efficiently by using a single event listener to manage all similar events on child elements by leveraging event bubbling.
3. **What is a promise?**
   * A promise is an object representing the eventual completion or failure of an asynchronous operation and its resulting value.
4. **How do you create a promise?**
   * Example: **let promise = new Promise((resolve, reject) => { /\* ... \*/ });**.
5. **What is async and await?**
   * **async** and **await** are used to write asynchronous code that appears synchronous. **async** declares a function as asynchronous, and **await** pauses execution until a promise is resolved.
6. **What is a callback function?**
   * A callback function is a function passed into another function as an argument to be executed later.
7. **What is the difference between null and undefined?**
   * **undefined** means a variable has been declared but has not yet been assigned a value. **null** is an assignment value that represents no value or no object.
8. **What are higher-order functions?**
   * Higher-order functions are functions that take other functions as arguments or return functions as their result.
9. **What is the spread operator?**
   * The spread operator (**...**) allows an iterable (like an array) to be expanded in places where zero or more arguments or elements are expected.
10. **What is destructuring?**
    * Destructuring is a syntax that allows you to unpack values from arrays or properties from objects into distinct variables.

**Advanced Questions**

1. **What is the event loop?**
   * The event loop is a mechanism that allows JavaScript to perform non-blocking operations by offloading operations to the system kernel whenever possible.
2. **What is the difference between synchronous and asynchronous code?**
   * Synchronous code is executed sequentially, whereas asynchronous code is executed without waiting for previous operations to complete.
3. **What is hoisting?**
   * Hoisting is JavaScript's default behavior of moving declarations to the top of the current scope.
4. **What are modules in JavaScript?**
   * Modules are reusable pieces of code that can be exported from one program and imported for use in another program.
5. **How do you create and use modules in JavaScript?**
   * Use the **export** keyword to export functions or variables, and **import** to use them in another file. Example:

// module.js

export const pi = 3.14;

// main.js

import { pi } from './module.js';

What is localStorage?

localStorage is a type of web storage that allows JavaScript sites and apps to store and access data right in the browser with no expiration time.

What is sessionStorage?

sessionStorage is similar to localStorage, but the data is only available for the duration of the page session.

What are Web Workers?

Web Workers allow you to run JavaScript scripts in the background, independently of any user interface scripts.

What is the difference between for...of and for...in?

for...of iterates over iterable objects (like arrays, strings), whereas for...in iterates over the properties of an object.

What is the purpose of Array.prototype.map()?

map() creates a new array populated with the results of calling a provided function on every element in the calling array.

These questions cover a wide range of topics from basic syntax and concepts to more advanced features and functionalities in JavaScript.

HTML question

1) What is HTML?

HTML stands for Hyper Text Markup Language. It is a language of World Wide Web. It is a standard text formatting language which is used to create and display pages on the Web. It makes the text more interactive and dynamic. It can turn text into images, tables, links.[More details.](https://www.javatpoint.com/what-is-html)

2) What are Tags?

HTML tags are composed of three things: an opening tag, content and ending tag. Some tags are unclosed tags.

HTML documents contain two things:

* content, and
* tags

When a web browser reads an HTML document, the browser reads it from top to bottom and left to right. HTML tags are used to create HTML documents and render their properties. Each HTML tags have different properties.

Syntax

1. **<tag>** content **</tag>**

Content is placed between tags to display data on the web page.

[More details.](https://www.javatpoint.com/html-tags)

3) Do all HTML tags have an end tag?

No. There are some HTML tags that don't need a closing tag. For example: <image> tag, <br> tag. [More details.](https://www.javatpoint.com/html-tags)

4) What is formatting in HTML?

The HTML formatting is a process of format the text for a better look and feel. It uses different tags to make text bold, italicized, underlined. [More details.](https://www.javatpoint.com/html-formatting)

5) How many types of heading does an HTML contain?

The HTML contains six types of headings which are defined with the <h1> to <h6> tags. Each type of heading tag displays different text size from another. So, <h1> is the largest heading tag and <h6> is the smallest one. For example:

1. **<h1>**Heading no. 1**</h1>**
2. **<h2>**Heading no. 2**</h2>**
3. **<h3>**Heading no. 3**</h3>**
4. **<h4>**Heading no. 4**</h4>**
5. **<h5>**Heading no. 5**</h5>**
6. **<h6>**Heading no. 6**</h6>**

[More details.](https://www.javatpoint.com/html-heading)

6) How to create a hyperlink in HTML?

The HTML provides an anchor tag to create a hyperlink that links one page to another page. These tags can appear in any of the following ways:

* Unvisited link - It is displayed, underlined and blue.
* Visited link - It is displayed, underlined and purple.
* Active link - It is displayed, underlined and red.

[More details.](https://www.javatpoint.com/html-anchor)

7) Which HTML tag is used to display the data in the tabular form?

The **HTML table tag** is used to display data in tabular form (row \* column). It also manages the layout of the page, e.g., header section, navigation bar, body content, footer section. Here is the list of tags used while displaying the data in the tabular form:

|  |  |
| --- | --- |
| **Tag** | **Description** |
| <table> | It defines a table. |
| <tr> | It defines a row in a table. |
| <th> | It defines a header cell in a table. |
| <td> | It defines a cell in a table. |
| <caption> | It defines the table caption. |
| <colgroup> | It specifies a group of one or more columns in a table for formatting. |
| <col> | It is used with <colgroup> element to specify column properties for each column. |
| <tbody> | It is used to group the body content in a table. |
| <thead> | It is used to group the header content in a table. |
| <tfooter> | It is used to group the footer content in a table. |

8) What are some common lists that are used when designing a page?

There are many common lists which are used to design a page. You can choose any or a combination of the following list types:

* Ordered list - The ordered list displays elements in numbered format. It is represented by <ol> tag.
* Unordered list - The unordered list displays elements in bulleted format. It is represented by <ul> tag.
* Definition list - The definition list displays elements in definition form like in dictionary. The <dl>, <dt> and <dd> tags are used to define description list.

[More details.](https://www.javatpoint.com/html-lists)

9) What is the difference between HTML elements and tags?

HTML elements communicate to the browser to render text. When the elements are enclosed by brackets <>, they form HTML tags. Most of the time, tags come in a pair and surround content.

10) What is semantic HTML?

Semantic HTML is a coding style. It is the use of HTML markup to reinforce the semantics or meaning of the content. For example: In semantic HTML <b> </b> tag is not used for bold statement as well as <i> </i> tag is used for italic. Instead of these we use <strong></strong> and <em></em> tags.

11) What is an image map?

Image map facilitates you to link many different web pages using a single image. It is represented by <map> tag. You can define shapes in images that you want to make part of an image mapping.

12) How to insert a copyright symbol on a browser page?

You can insert a copyright symbol by using &copy; or &#169; in an HTML file.

13) How to create a nested webpage in HTML?

The HTML iframe tag is used to display a nested webpage. In other words, it represents a webpage within a webpage. The HTML <iframe> tag defines an inline frame. For example:

1. <!DOCTYPE html**>**
2. **<html>**
3. **<body>**
4. **<h2>**HTML Iframes example**</h2>**
5. **<p>**Use the height and width attributes to specify the size of the iframe:**</p>**
6. **<iframe** src="https://www.javatpoint.com/" height="300" width="400"**></iframe>**
7. **</body>**
8. **</html>**

14) How do you keep list elements straight in an HTML file?

You can keep the list elements straight by using indents.

15) Does a hyperlink only apply to text?

No, you can use hyperlinks on text and images both. The HTML anchor tag defines a hyperlink that links one page to another page. The "href" attribute is the most important attribute of the HTML anchor tag.

Syntax

1. **<a** href = "..........."**>** Link Text **</a>**

[More details.](https://www.javatpoint.com/html-anchor)

16) What is a style sheet?

A style sheet is used to build a consistent, transportable, and well-designed style template. You can add these templates on several different web pages. It describes the look and formatting of a document written in markup language.

17) Can you create a multi-colored text on a web page?

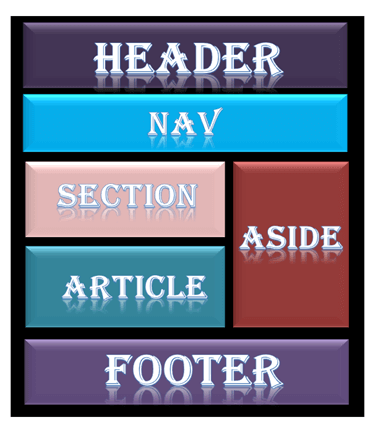
Yes. To create a multicolor text on a web page you can use <font color ="color"> </font> for the specific texts you want to color.

18) Is it possible to change the color of the bullet?

The color of the bullet is always the color of the first text of the list. So, if you want to change the color of the bullet, you must change the color of the text.

19) Explain the layout of HTML?

HTML layout specifies a way in which the web page is arranged.



Every website has a specific layout to display content in a specific manner.

Following are different HTML5 elements which are used to define the different parts of a webpage.

* <header>: It is used to define a header for a document or a section.
* <nav>: It is used to define a container for navigation links
* <section>: It is used to define a section in a document
* <article>: It is used to define an independent, self-contained article
* <aside>: It is used to define content aside from the content (like a sidebar)
* <footer>: It is used to define a footer for a document or a section

20) What is a marquee?

Marquee is used to put the scrolling text on a web page. It scrolls the image or text up, down, left or right automatically. You should put the text which you want to scroll within the <marquee>......</marquee> tag. [More details.](https://www.javatpoint.com/marquee-html)

21) How many tags can be used to separate a section of texts?

Three tags are used to separate the texts.

* <br> tag - Usually <br> tag is used to separate the line of text. It breaks the current line and conveys the flow to the next line
* <p> tag - The <p> tag contains the text in the form of a new paragraph.
* <blockquote> tag - It is used to define a large quoted section. If you have a large quotation, then put the entire text within <blockquote>.............</blockquote> tag.

22) How to make a picture of a background image of a web page?

To make a picture a background image on a web page, you should put the following tag code after the </head> tag.

1. **<body** background = "image.gif"**>**

Here, replace the "image.gif" with the name of your image file which you want to display on your web page.

23) What are empty elements?

HTML elements with no content are called empty elements. For example: <br>, <hr> etc.

24) What is the use of a span tag? Give one example.

The span tag is used for following things:

* For adding color on text
* For adding background on text
* Highlight any color text

**Example:**

1. **<p>**
2. **<span** style="color:#ffffff;"**>**
3. In this page we use span.
4. **</span>**
5. **</p>**

25) What is the use of an iframe tag?

An iframe is used to display a web page within a web page.

**Syntax:**

1. **<iframe** src="URL"**></iframe>**

**Example:**

1. **<iframe** src="demo\_iframe.html" width="200px" height="200px"**></iframe>**

**Target to a link:**

1. **<iframe** src="http://www.javatpoint.com" name="iframe\_a"**></iframe>**

26) What are the entities in HTML?

The HTML character entities are used as a replacement for reserved characters in HTML. You can also replace characters that are not present on your keyboard by entities. These characters are replaced because some characters are reserved in HTML.

27) Why is a URL encoded in HTML?

An URL is encoded to convert non-ASCII characters into a format that can be used over the Internet because a URL is sent over the Internet by using the ASCII character-set only. If a URL contains characters outside the ASCII set, the URL has to be converted. The non-ASCII characters are replaced with a "%" followed by hexadecimal digits.

28) Does a <!DOCTYPE html> tag is a HTML tag?

No, the <!DOCTYPE html> declaration is not an HTML tag. There are many type of HTML e.g. HTML 4.01 Strict, HTML 4.01 Transitional, HTML 4.01 Frameset, XHTML 1.0 Strict, XHTML 1.0 Transitional, XHTML 1.0 Frameset, XHTML 1.1 etc. So, <!DOCTYPE html> is used to instruct the web browser about the HTML page.

HTML5 Interview Questions

Let's see a list of top HTML5 interview questions and answers.

29) What is the canvas element in HTML5?

The <canvas> element is a container that is used to draw graphics on the web page using scripting language like JavaScript. It allows for dynamic and scriptable rendering of 2D shapes and bitmap images. There are several methods in canvas to draw paths, boxes, circles, text and add images. For Example:

1. **<canvas** id="myCanvas1" width="300" height="100" style="border:2px solid;"**>**
2. Your browser does not support the HTML5 canvas tag.
3. **</canvas>**

[More details.](https://www.javatpoint.com/html-canvas)

30) What is SVG?

HTML SVG is used to describe the two-dimensional vector and vector/raster graphics. SVG images and their behaviors are defined in XML text files. So as XML files, you can create and edit an SVG image with the text editor. It is mostly used for vector type diagrams like pie charts, 2-Dimensional graphs in an X, Y coordinate system.

1. **<svg** width="100" height="100"**>**
2. **<circle** cx="50" cy="50" r="40" stroke="yellow" stroke-width="4" fill="red" **/>**
3. **</svg>**

[More details.](https://www.javatpoint.com/html-svg)

31) What are the different new form element types in HTML 5?

Following is a list of 10 frequently used new elements in HTML 5:

* Color
* Date
* Datetime-local
* Email
* Time
* Url
* Range
* Telephone
* Number
* Search

32) Is there any need to change the web browsers to support HTML5?

No. Almost all browsers (updated versions) support HTML 5. For example Chrome, Firefox, Opera, Safari, IE.

33) Which type of video formats are supported by HTML5?

HTML 5 supports three types of video format:

* mp4
* WebM
* Ogg

[More details.](https://www.javatpoint.com/html-video)

34) Is audio tag supported in HTML 5?

Yes. It is used to add sound or music files on the web page. There are three supported file formats for HTML 5 audio tag.

1. mp3
2. WAV
3. Ogg

Let's see the code to play mp3 file using HTML audio tag.

1. **<audio** controls**>**
2. **<source** src="koyal.mp3" type="audio/mpeg"**>**
3. Your browser does not support the html audio tag.
4. **</audio>**

Instead of koyal.mp3, you can pass any mp3 file name.

[More details.](https://www.javatpoint.com/html-audio)

35) What is the difference between progress and meter tag?

The progress tag is used to represent the progress of the task only while the meter tag is used to measure data within a given range. [More details.](https://www.javatpoint.com/html-progress-tag)

36) What is the use of figure tag in HTML 5?

The figure tag is used to add a photo in the document on the web page. It is used to handle the group of diagrams, photos, code listing with some embedded content.

1. **<p>**The Taj Mahal is widely recognized as "the jewel of Muslim art in India and one of the universally admired masterpieces of the world's heritage."**</p>**
2. **<figure>**
3. **<img** src="htmlpages/images/tajmahal.jpg" alt="Taj Mahal"**/>**
4. **</figure>**

[More details.](https://www.javatpoint.com/html-figure-tag)

37) What is the use of figcaption tag in HTML 5?

The <figcaption> element is used to provide a caption to an image. It is an optional tag and can appear before or after the content within the <figure> tag. The <figcaption> element is used with <figure> element and it can be placed as the first or last child of the <figure> element.

1. **<figure>**
2. **<img** src="htmlpages/images/tajmahal.jpg" alt="Taj Mahal"**/>**
3. **<figcaption>**Fig.1.1 - A front view of the great Taj Mahal in Agra.**</figcaption>**
4. **</figure>**

[More details.](https://www.javatpoint.com/html-figcaption-tag)

38) What is button tag?

The button tag is used in HTML 5. It is used to create a clickable button within the HTML form on the web page. It is generally used to create a "submit" or "reset" button. Let's see the code to display the button.

1. **<button** name="button" type="button"**>**Click Here**</button>**

[More details.](https://www.javatpoint.com/html-button-tag)

39) What is the use of details and summary tag?

The details tag is used to specify some additional details on the web page. It can be viewed or hidden on demand. The summary tag is used with details tag. [More details.](https://www.javatpoint.com/html-details-tag)

40) What is datalist tag?

The HTML 5 datalist tag provides an autocomplete feature on the form element. It facilitates users to choose the predefined options to the users to select data.

1. **<label>**
2. Enter your favorite cricket player: Press any character**<br** **/>**
3. **<input** type="text" id="favCktPlayer" list="CktPlayers"**>**
4. **<datalist** id="CktPlayers"**>**
5. **<option** value="Sachin Tendulkar"**>**
6. **<option** value="Brian Lara"**>**
7. **<option** value="Jacques Kallis"**>**
8. **<option** value="Ricky Ponting"**>**
9. **<option** value="Rahul Dravid"**>**
10. **<option** value="Shane Warne"**>**
11. **<option** value="Rohit Sharma"**>**
12. **<option** value="Donald Bradman"**>**
13. **<option** value="Saurav Ganguly "**>**
14. **<option** value="AB diVilliers"**>**
15. **<option** value="Mahendra Singh Dhoni"**>**
16. **<option** value="Adam Gilchrist"**>**
17. **</datalist>**
18. **</label>**

[More details.](https://www.javatpoint.com/html-datalist-tag)

41) How are tags migrated from HTML4 to HTML5?

|  |  |  |
| --- | --- | --- |
| **No.** | **Typical HTML4** | **Typical HTML5** |
| 1) | <div id="header"> | <header> |
| 2) | <div id="menu"> | <nav> |
| 3) | <div id="content"> | <section> |
| 4) | <div id="post"> | <article> |
| 5) | <div id="footer"> | <footer> |

Header and Footer Example

**HTML 4 Header and Footer:**

1. **<div** id="header"**>**
2. **<h1>**Monday Times**</h1>**
3. **</div>**
4. .
5. .
6. .
7. **<div** id="footer"**>**
8. **<p>**&copy; JavaTpoint. All rights reserved.**</p>**
9. **</div>**

**HTML 5 Header and Footer:**

1. **<header>**
2. **<h1>**Monday Times**</h1>**
3. **</header>**
4. .
5. .
6. .
7. **<footer>**
8. **<p>**© JavaTpoint. All rights reserved.**</p>**
9. **</footer>**

Menu Example

**HTML 4 Menu:**

1. **<div** id="menu"**>**
2. **<ul>**
3. **<li>**News**</li>**
4. **<li>**Sports**</li>**
5. **<li>**Weather**</li>**
6. **</ul>**
7. **</div>**

**HTML 5 Menu:**

1. **<nav>**
2. **<ul>**
3. **<li>**News**</li>**
4. **<li>**Sports**</li>**
5. **<li>**Weather**</li>**
6. **</ul>**
7. **</nav>**

42) If I do not put <!DOCTYPE html> will HTML 5 work?

No, the browser will not be able to identify that it is an HTML document and HTML 5 tags do not function properly..

43) What is the use of the required attribute in HTML5?

It forces a user to fill text on the text field or text area before submitting the form. It is used for form validation.

**Example:**

1. Name: **<input** type="text" name="name" required**>**

44) What are the new <input> types for form validation in HTML5?

The new input types for form validation are email, URL, number, tel, and date.

**Example:**

1. **<input** type="email"**>**

Sure! Here is an example of a coding round task for creating an e-commerce website using PHP and MySQL. This task will cover basic functionalities like user registration, login, product listing, and adding products to the cart.

Task: E-commerce Website

Requirements

User Registration and Login

Create a user registration form.

Create a user login form.

Securely store user credentials in the database.

Product Management

Create a form to add new products.

Display a list of products.

Shopping Cart

Allow users to add products to their cart.

Display the cart with the selected products.

Step-by-Step Guide

Step 1: Setting Up the Database

Create a MySQL database named ecommerce.

sql

Copy code

CREATE DATABASE ecommerce;

USE ecommerce;

-- Users table

CREATE TABLE users (

id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(50) NOT NULL UNIQUE,

password VARCHAR(255) NOT NULL,

email VARCHAR(100) NOT NULL

);

-- Products table

CREATE TABLE products (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100) NOT NULL,

description TEXT,

price DECIMAL(10, 2) NOT NULL

);

-- Cart table

CREATE TABLE cart (

id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT,

product\_id INT,

quantity INT,

FOREIGN KEY (user\_id) REFERENCES users(id),

FOREIGN KEY (product\_id) REFERENCES products(id)

);

Step 2: User Registration

Create a register.php file for user registration.

php

Copy code

<?php

if ($\_SERVER['REQUEST\_METHOD'] == 'POST') {

$username = $\_POST['username'];

$password = password\_hash($\_POST['password'], PASSWORD\_BCRYPT);

$email = $\_POST['email'];

$conn = new mysqli('localhost', 'root', '', 'ecommerce');

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

$stmt = $conn->prepare("INSERT INTO users (username, password, email) VALUES (?, ?, ?)");

$stmt->bind\_param("sss", $username, $password, $email);

if ($stmt->execute()) {

echo "Registration successful!";

} else {

echo "Error: " . $stmt->error;

}

$stmt->close();

$conn->close();

}

?>

<form method="post" action="">

Username: <input type="text" name="username" required><br>

Password: <input type="password" name="password" required><br>

Email: <input type="email" name="email" required><br>

<input type="submit" value="Register">

</form>

Step 3: User Login

Create a login.php file for user login.

php

Copy code

<?php

session\_start();

if ($\_SERVER['REQUEST\_METHOD'] == 'POST') {

$username = $\_POST['username'];

$password = $\_POST['password'];

$conn = new mysqli('localhost', 'root', '', 'ecommerce');

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

$stmt = $conn->prepare("SELECT id, password FROM users WHERE username = ?");

$stmt->bind\_param("s", $username);

$stmt->execute();

$stmt->store\_result();

$stmt->bind\_result($id, $hashed\_password);

if ($stmt->fetch() && password\_verify($password, $hashed\_password)) {

$\_SESSION['user\_id'] = $id;

echo "Login successful!";

} else {

echo "Invalid username or password.";

}

$stmt->close();

$conn->close();

}

?>

<form method="post" action="">

Username: <input type="text" name="username" required><br>

Password: <input type="password" name="password" required><br>

<input type="submit" value="Login">

</form>

Step 4: Add Products

Create an add\_product.php file for adding products.

php

Copy code

<?php

if ($\_SERVER['REQUEST\_METHOD'] == 'POST') {

$name = $\_POST['name'];

$description = $\_POST['description'];

$price = $\_POST['price'];

$conn = new mysqli('localhost', 'root', '', 'ecommerce');

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

$stmt = $conn->prepare("INSERT INTO products (name, description, price) VALUES (?, ?, ?)");

$stmt->bind\_param("ssd", $name, $description, $price);

if ($stmt->execute()) {

echo "Product added successfully!";

} else {

echo "Error: " . $stmt->error;

}

$stmt->close();

$conn->close();

}

?>

<form method="post" action="">

Name: <input type="text" name="name" required><br>

Description: <textarea name="description"></textarea><br>

Price: <input type="text" name="price" required><br>

<input type="submit" value="Add Product">

</form>

Step 5: List Products

Create a list\_products.php file to display the products.

php

Copy code

<?php

$conn = new mysqli('localhost', 'root', '', 'ecommerce');

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

$result = $conn->query("SELECT id, name, description, price FROM products");

if ($result->num\_rows > 0) {

while ($row = $result->fetch\_assoc()) {

echo "Product ID: " . $row['id'] . "<br>";

echo "Name: " . $row['name'] . "<br>";

echo "Description: " . $row['description'] . "<br>";

echo "Price: $" . $row['price'] . "<br>";

echo "<a href='add\_to\_cart.php?id=" . $row['id'] . "'>Add to Cart</a><br><br>";

}

} else {

echo "No products found.";

}

$conn->close();

?>

Step 6: Add to Cart

Create an add\_to\_cart.php file to add products to the cart.

php

Copy code

<?php

session\_start();

if (!isset($\_SESSION['user\_id'])) {

die("You must be logged in to add products to the cart.");

}

if (isset($\_GET['id'])) {

$user\_id = $\_SESSION['user\_id'];

$product\_id = $\_GET['id'];

$quantity = 1; // default quantity

$conn = new mysqli('localhost', 'root', '', 'ecommerce');

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

$stmt = $conn->prepare("INSERT INTO cart (user\_id, product\_id, quantity) VALUES (?, ?, ?)");

$stmt->bind\_param("iii", $user\_id, $product\_id, $quantity);

if ($stmt->execute()) {

echo "Product added to cart!";

} else {

echo "Error: " . $stmt->error;

}

$stmt->close();

$conn->close();

}

?>

<a href="list\_products.php">Back to Products</a>

Step 7: Display Cart

Create a view\_cart.php file to display the cart contents.

php

Copy code

<?php

session\_start();

if (!isset($\_SESSION['user\_id'])) {

die("You must be logged in to view the cart.");

}

$user\_id = $\_SESSION['user\_id'];

$conn = new mysqli('localhost', 'root', '', 'ecommerce');

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

$stmt = $conn->prepare("SELECT p.name, p.price, c.quantity

FROM cart c

JOIN products p ON c.product\_id = p.id

WHERE c.user\_id = ?");

$stmt->bind\_param("i", $user\_id);

$stmt->execute();

$result = $stmt->get\_result();

if ($result->num\_rows > 0) {

while ($row = $result->fetch\_assoc()) {

echo "Product: " . $row['name'] . "<br>";

echo "Price: $" . $row['price'] . "<br>";

echo "Quantity: " . $row['quantity'] . "<br><br>";

}

} else {

echo "Your cart is empty.";

}

$stmt->close();

$conn->close();

?>

This example provides a basic structure for an e-commerce website. You can further enhance it by adding more features like user profiles, order management, payment integration, product categories, search functionality, etc.

Bottom of Form