



S2 Labs

Top 30 LWC

Interview Question and Answer



1. What is LWC?

Lightning web components, better known as LWC, are introduced by Salesforce. It is the modern framework that builds user interfaces by using standard web technologies like JavaScript and HTML. It helps developers create reusable, effective, and efficient components that can improve the performance of Salesforce applications. LWC also provides an enhanced development experience, better performance, and great compatibility across multiple browsers and different devices.

2. How is Aura different from Lightning Web Components?

Aura and LWC (Lightning Web Components) are both frameworks used for building components in the Salesforce ecosystem, but they have some major differences:

- Aura is a proprietary framework, while Lightning web components use standard web technologies such as JavaScript, HTML, and CSS.
- LWC performs better than Aura due to its native web standards and effective rendering.
- Aura is proprietary, so developers must have adequate experience before using this framework. On the other hand, LWC is easy to learn and helps developers become familiar with standard web technologies.

3. What do you mean by Salesforce Lightning?

Salesforce Lightning is a component-based framework useful for Salesforce application development. Its primary objective is to ease the development process for developers who lack adequate programming experience.

4. What is the use of Lightning Components Framework?

In layman's terms, the Lightning component framework is a UI framework helpful for developing dynamic applications for desktop and mobile users.

5. Mention some Lightning Salesforce Components.

There are multiple components in Salesforce Lightning, such as:

1. Lightning Button

2. Lightning Input
3. Lightning Combo Box
4. Lightning Exchange
5. Lightning Design System

6. What is renderedCallback() in LWC?

RenderedCallback() is the function or method in LWC that is useful for carrying out the logic once the component completes its rendering phase. This lifecycle hook is important for performing actions required after the component gets rendered, like manipulating the DOM (Document Object Model) and initializing the third-party libraries.

7. What Is the use of @Wire in Lightning Web Components?

The @wire decorator is used in LWC to associate a property or method with a Salesforce data source. It automatically manages data collecting, error handling, and state management, making it easier to integrate Salesforce data into components. The @wire decorator is compatible with Apex methods and Salesforce's built-in wire adapters.

8. Name the tools present in the Salesforce Lightning.

Salesforce Lightning includes the Lightning Component Framework, Lightning App Builder, Lightning Process Builder, Lightning Design System, Lightning Component Library, and Lightning Connect, which are useful tools.

9. What is Lightning Process Builder?

Lightning Process Builder is a visual tool for automating business processes and workflows within Salesforce. It offers a user-friendly interface for creating complex processes with a point-and-click method, allowing you to perform operations like changing records, sending emails, and running Apex programs without having to write any code.

10. What is the use of SFDX?

SFDX stands for "Salesforce Developer Experience." This tool is useful because it improves developers' experience working on the Salesforce platform.

11. What is meant by cacheable = True Annotations?

When you set `cacheable = true`, the function can only get data and cannot perform any data manipulation language (DML) actions. This is used to improve component performance by displaying cached data from client-side storage instead of waiting for a server trip. To refresh obsolete Apex data, we must utilize `refreshApex` because LDS (Lightning Data Service) does not handle data from Apex (in the case of a wired service).

12. When is the Property or Wire Method called in the Component Lifecycle?

Wire methods or properties are called when any reactive parameter gets changed, and the component is initialized. The `@wire` decorator ensures automated updates and data fetching.

13. What are the Lifecycle Hooks in LWC?

A lifecycle hook is a callback method invoked at a specified stage of a component instance's lifecycle. LWC supports the following hooks.

- **Constructor:** Called when the component has been generated.
- **ConnectedCallback:** This callback is triggered when the component is added to the document.
- **RenderedCallback:** Called after each render of the component. This lifecycle hook is particular to Lightning Web Components and is not included in the definition of HTML custom elements. It flows from child to parent and is defined in `LightningElement` rather than `HTMLElement`.
- **DisconnectedCallback:** This function is required when an component is deleted from a document.
- **ErrorCallback:** This method is called whenever a descendant component throws an error. The error argument is the JavaScript error object, and the stack argument is a string. This lifecycle hook is particular to LWC and is not included in the definition of HTML custom elements.

14. Do quick actions support the LWC Components?

Yes, quick actions support the LWC. Individuals can use lightning web components to generate custom quick actions. However, these components only help on the record pages.

15. What do you mean by a Promise in Async Transactions? Also, Name its Different Stages.

A promise is an object returned by an asynchronous transaction that notifies you when the transaction has finished or encountered an error. For example, when you make an urgent call to an apex, it produces a promise object. Depending on the returned object, the execution will either enter the 'then' block, indicating a successful transaction, or the 'catch' block, indicating a failed transaction.

The stages of a promise are:

- Pending: Awaiting a result.
- Fulfilled: The promise is successful.
- Rejected: The promise has failed.

LWC Interview Questions for Experienced Professionals

As an LWC developer, having some prior experience before switching to a new job in the industry will help you out. But reading these advanced Salesforce Lightning interview questions will prepare you for the next interview round.

16. What are Web Components? Is the Process of LWC Based on the Web Components?

Web Components are a collection of web platform APIs that allow developers to create reusable, contained HTML tags for custom elements. They consist of the Shadow DOM, Custom Elements, and HTML Templates. LWC is built on Web Components and uses standard APIs to develop fast, flexible, and reusable elements.

17. What Is the Difference between Connectedcallback and Renderedcallback In LWC?

Lifecycle hooks are unique methods in LWC that help developers run code at defined stages of a component's lifecycle. ConnectedCallback and renderedCallback are two of the most used lifecycle hooks. These hooks allow developers to run code when a component is placed into the DOM and after it has been rendered. Understanding the distinctions between these hooks is essential for efficiently handling component initialization and updates.

18. Are Callback Functions Asynchronous or Synchronous?

Callback functions can be synchronous and asynchronous, but it depends on the context where they are being passed. For example, callbacks in `setTimeout()` or `fetch()` are asynchronous, while callbacks in array methods like `forEach()` are synchronous.

19. What do you mean by Callback Hell?

In simple terms, callback hell refers to the situation where multiple stacked callback functions result in a complex code structure. It frequently happens in asynchronous programming when callbacks are utilized extensively, resulting in highly nested code that is difficult to maintain and debug.

20. What are Decorators in LWC?

There are three decorators for the LWC programming model that add functionality to the functions or properties.

1. `@track`
2. `@api`
3. `@wire`

21. What is the difference between var and let in the Java Script?

'var' is the function scoped variable which can be redeclared. On the contrary, 'let' is the block-scoped variable that cannot be redeclared within the same scope.

'var' declared variables are hoisted on the top of their scope. On the other hand, 'let' variables offer better scoping rules and eliminate the issues related to variable hoisting.

22. What is LMS?

The acronym LMS stands for Lightning Message Service. It enables smooth communication among Lightning Web Components, Aura Components, and Visualforce sites. LMS allows for message broadcasting and listening across various areas of the program, facilitating inter-component communication.

23. Are there any Application Events in LWC?

No, LWC does not support the application events used by Aura Components. Rather, LWC communicates with components via a regular DOM event model. Custom Events can be dispatched and listened to, bringing the framework in line with modern web development methods.

24. What do you mean by String Interpolation?

String interpolation refers to the process of embedding expressions or variables within a string. This is commonly done in JavaScript with template literals, which are indicated by backticks (`).

25. When we face the Error of “Cannot Assign to Read Only Property” in the Lightning Web Components?

This error occurs when you attempt to assign a value to a read-only property. In LWC, it can occur if you attempt to directly change a public property passed down from a parent component or if you attempt to edit the properties of an immutable object.

26. How can you pass data between the LWC?

Passing data between components in LWC is a common requirement to create dynamic and interactive applications. There are multiple ways to do this process, but it depends on the relationship between the components. There are some common methods through which data can be passed between the Lightning Web Components, which include:

Properties: Parent components can send data to child components using public properties marked with `@api`.

Events: Child components can communicate with parent components by triggering custom events that the parent components can manage.

27. How can you Communicate from Child to Parent Component in LWC?

In LWC, communication between a child and a parent component happens through custom events. The child component uses the `CustomEvent` constructor to dispatch an event, which is then received and handled by the parent component.

28. How can you Render Elements in LWC Conditionally?

LWC achieves conditional rendering with template directives like 'if: true' and 'if: false'. These directives include or exclude elements of the template depending on the value of a Boolean property.

29. What is the Difference between "===" and "=="?

Both == and === are used to compare two variables. However, == does not consider data type and performs type coercion, which means that the interpreter attempts to transform data types to match the values automatically. On the other hand, === always returns false if the data types are not the same.

For example:

2 == "2" will return True (type coercion occurs).

2 === "2" returns False (no type coercion).

30. How do you Handle Asynchronous Operations in LWC?

Handling the asynchronous operations in the Lightning Web Components is essential to manage tasks like processing the user inputs, performing background computations, and fetching data from the server. These operations ensure that UI responds when these tasks are completed.