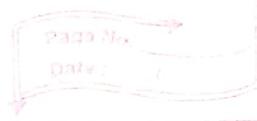


LWC Topic

- Q-1) what is LWC ?
- ① LWC stands for Lightning web Component.
 - ② LWC is an implementation of the W3C's web component standards.
 - ③ it supports parts of web component that works in browsers and adds parts supported by Salesforce as well.

- ④ Quick component development because developer has to use only HTML, CSS and JavaScript.

- Q-2) Lightning web Components files.

- ① `lightning.html` contains a base framework.
- ② `test.js` is a functional test file.
- ③ `test.css` and `style.css` are used for styling.
- ④ `testwix-testlet.xml` is used for end-to-end testing with the help of `test.html`.

- Q-3) what is LWC module?

- ① LWC uses modules to bundle core functionality and make it accessible to the JavaScript in your component file.
- ② The core module for lightning web component is LWC.
- ③ Begin the module with the import statement and specify the functionality of the module that your component uses.
- ④ Example: `import { LightningElement } from 'lwc';`
- ⑤ The import statement indicates the JavaScript uses the Lightning Element functionality from the LWC module.

Q-4) what is use of yml file?

→ ① This file ~~and defines~~ component's configuration
information where developer can use
the lightning web components

② <isExposed> true </isExposed>

③ <targets>

<target> lightning__HomePage </target>

<target> lightning__RecordPage </target>

</targets>

Q-5) Can Aura Component contain lightning web
Component?

→ Yes

Q-6) Can lightning web Component contain
Aura Component?

→ No

Q-7) can a lightning web component call another
lightning web component?

→ Yes

Q-8) Camel Case vs kebab Case?

→ ① We match web standard whenever
possible. The ~~HTML standard~~ requires
that custom element names contain a hyphen.
Since all LWC have a namespace that's separate
from the folder name by a hyphen.
Component names meet the ~~HTML~~ standard.

③ For example, the markup for the lightning web component with the folder name Widget in the default namespace is: `<c-widget>`

④ However, the Salesforce platform doesn't allow hyphens in the component folder or file names. What if a component's name has more than one word, like "MyComponent"? You can't name the folder and file my-component, but we do have a handy solution.

⑤ Use camel case to name your component like myComponent. Camel case Component folder names map to kebab-case for markup. In markup, to reference a component with the folder name myComponent, use `<c-my-component>`

Q) what is Decoration?

→ ① Decorators are often used in JavaScript to modify the behavior of a property or function.

Example → `method abc def`

→ ② `@capital` → defining

capital of a country

→ ③ `@track` → marks method for

change effect whenever

→ ④ `@wire` → mark method to

inject a dependency object into each

Q) what is the use of `@capital` decorator?

- ① Marks a field / property as public
- ② HTML markup can access the component's public properties.
- ③ All public properties are reactive.
Reactive means the framework observes the property for change. When property changes value then the framework reacts and renders the component.

Q-1) what is the use of `@track` Decorator ?

- ① Observes changes to the properties of an object or to the elements of an array.
- ② Framework renders the component when changes occurs.

Q-2) what is the use of `@wire` decorator ?

- ① it provides a way to get and bind data from a salesforce org.

Q-3) Conditional Rendering in HTML ?

```
<template if: true = {areDetailsVisible} >  
    This is true block!  
</template>  
<template if: false = {areDetailsVisible} >  
    This is false block!  
</template>
```

3 (true). spread operator

Q-14) Rendering a list in HTML

→ <template for="item" items="contacts">

<li key="contactId">

 <div> { contact.Name } , { contact.Title } </div>

</template>

Q-15) How to write expression in Component

→ <template> { item.itemName } </template>

 { itemName } got converted to

 </template>

français with numbers displayed.

II todoItem.js

```
import { LightningElement, api } from 'lwc';
export default class todoItem extends
  lightningElement {
  @api itemName;
}
```

Q-16) How to call Controller function in component

→ HTML

```
<lightning-input type="checkbox" label="Show details"
  checked="checked" onchange="handleChange"></lightning-input>
```

JS

```
handleChange(event) {
  this.areDetailsVisible = event.target.checked;
```

Q-17) @AuraEnabled (cacheable = true)
→ AuraEnabled annotation exposes the
method to lightning components and
caches the returned list on the client.

Q-18) what is Lifecycle Hooks?
① lightning web components provides
methods that allow you to "hook"
your code up to critical events in a
component's lifecycle.
② These events include when a component is

- ① Created - constructor()
- ② Added to the Dom - connectedCallback()
- ③ Rendered in the browser - renderedCallback()
- ④ Encountering errors - errorCallback()
- ⑤ Removed from the Dom - disconnectedCallback()

Q-19) what is Lifecycle Flow?
① constructor called on parent
② public property value of parent updated
③ parent inserted into the Dom
④ connectedCallback() called on parent.
⑤ parent rendered
⑥ constructor called on child
⑦ public property value of child updated
⑧ child inserted into the Dom.
⑨ connectedCallback() called on child
⑩ child rendered
⑪ renderedCallback() called on child
⑫ renderedCallback() called on parent

(Q-20) What are the ways to work with Salesforce Lightning Components Built on Lightning Data Services?

- ① Use Basic Lightning Components Built on Lightning Data Services.
- ① lightning - record-form
- ② lightning - record-view-form
- ③ lightning - record - credit - off form

② Use Lightning Data Service via Adapters and Functions.

- ① lightning / UI/ API module
- ② Use Apex

(Q-21) Lighting Data Service does a lot of work.

- ① Lightning Data Service manages data for changes to a record and architecture in all the technologies built on it.

Whereas, data from Apex is not managed. You must refresh the data.

- ② Lightning Data Service does a lot of work to make code better well:

① Loads records in data progressively.

② Caches results on the client.

③ Invalidates cache entries when dependent Salesforce data and metadata changes.

- ④ Optimizes server calls by bulkifying the deduplicating requests.

bulkify no further combining them

breaks into smaller chunks based on

transact on paths (good for stock)

Q-22) Base Component Comparison

Feature	Lightning Record Form	Lightning Record View Form	Lightning Record Edit Form
Create Records	Yes	Yes	Yes
Edit Records	Yes	Yes	Yes
View Records	Yes	Yes	Yes
Read-Only Mode	Yes	Yes	Yes
Layout Types	Yes	Yes	Yes
Multi Column Layout	Yes	Yes	Yes
Custom Layout for Fields	Yes	Yes	Yes
Custom Rendering of Record Data	Yes	Yes	Yes

- Q-23) → Write Service
- ① The write service provisions an immutable stream of data to the component.
 - ② Each value in the stream is a newer version of the value that precedes it.
 - ③ Objects passed to a component are read-only.
 - ④ To mutate the data, a component should make a shallow copy of the objects it wants to mutate.

(C-24) Wire Service Syntax

```
import { adapterId } from 'adapterModule';
@wire (adapterId; adapterConfig)
propertyOrFunction;
```

- ① adapterId (Identifier) → The identifier of the wire adapter.
- ② adapterModule (String) → The identifier of the module that contains the wire adapter function, in the form of `moduleName`.
- ③ adapterConfig (Object) → A configuration Object specific to the wire adapter. Configuration Object property value can be either string or references to objects and fields imported from @salesforce/schema.

- ④ propertyOrFunction → A private property or function that receive the String of data from the wire service. If a property is decorated with `@wire`, the results are returned to the property's `data` property or `error` property. If a function is decorated with `@wire`, the results are returned in an object with a `data` property and an `error` property.

- Q-25) why to import References to Salesforce objects and fields?
- ① when you use a wire adapter in a lightning / ui API module. we strongly recommend importing references to objects and fields.
 - ② Salesforce verifies that the objects and fields exist, prevents objects and fields from being deleted, and cascades any renamed objects and fields into your component's source code.
 - ③ it also ensures that dependent objects and fields are included in change sets and packages.
 - ④ If a component isn't aware of which object's it's using, use strings instead of imported references.
use `getObjectType` to return the object's fields.
 - ⑤ All wire adapters in the lightning / ui API modules respect object CRUD rules, field-level security, and sharing.
 - ⑥ if a user doesn't have access to a field it isn't included in the response.

- Q-26) How to import References to Salesforce objects and fields?
- ① import `Position__object` from '@salesforce/schema/Position__c';
 - ② import `Account__object` from '@salesforce/schema/Account';

- ③ import POSITION_LEVEL_FIELD from '@salesforce/schema/Position.c_Level_c';
- ④ import ACCOUNT_NAME_FIELD from '@salesforce/schema/Account.Name';
- ⑤ import POSITION_HIRINGMANAGER_NAME_FIELD from '@salesforce/schema/Position.c_HiringManager__r.Name';
- ⑥ import ACCOUNT_OWNER_NAME_FIELD from '@salesforce/schema/Account.Owner.Name';

Q-27) How to get current record id in lightning web component?

→ Create a property named `recordId` and
any decorate it with `@api` decorator

Q-28) How can we deploy lightning web components
→ lightning components can be deployed
not like any other component using
change set, ant migration tool,
~~Grantset~~, ~~Copy & paste~~ others migration tool.

Q-29) Communicate with Events in LWC?

- ① Lightning web components dispatch standard Dom events.
- ② Components can also create and dispatch custom events.
- ③ You can use events to communicate up the component containment hierarchy.
- ④ Create and dispatch events in a component's JavaScript file.

- ① To Create an event, use the CustomEvent() constructor. To dispatch an event, call the EventTarget.dispatchEvent(Event() method).
- ② To listen on event use Component's HTML template. To handle events, define methods in the component's JavaScript class.

- Q-3) Use of Lightning Message Service?
- ① To communicate b/w components within within a single lightning page or across multiple pages, use Lightning Message Service. To communicate over a Lightning message channel.
- ② The advantage over pubsub module is that message channels aren't restricted to a single page.
- ③ Any component in a lightning Experience application that listens for events on a message channel updates when it receives a message.
- ④ It works b/w lightning web components, Aura Components, and VF Pages in any tab or in any pop-out window in Lightning Experience.

(Q-31) Use of pubsub module &

- ① To containers that don't support Lightning Messaging Service, use the pubsub module.
- ② In a publish-subscribe pattern, one component publishes an event and other component subscribes to receive and handle the event.
- ③ Every component that subscribes to the event receives the event.
- ④ The pubsub module reacts to events as soon as it receives a single publication.

(Q-32) Which components use Lightning web Components?

- ① Distributing Components on App Exchange
- ② Lightning web app > Builder
- ③ Flows
- ④ Experienced Builder
- ⑤ Utility Bar
- ⑥ Create Components for Outlook and Gmail integrations.
- ⑦ Quick Actions
- ⑧ Standard & Standard Aura Apps
- ⑨ Visualforce pages
- ⑩ Custom interface