How to Use LWC in a Screen Flow?

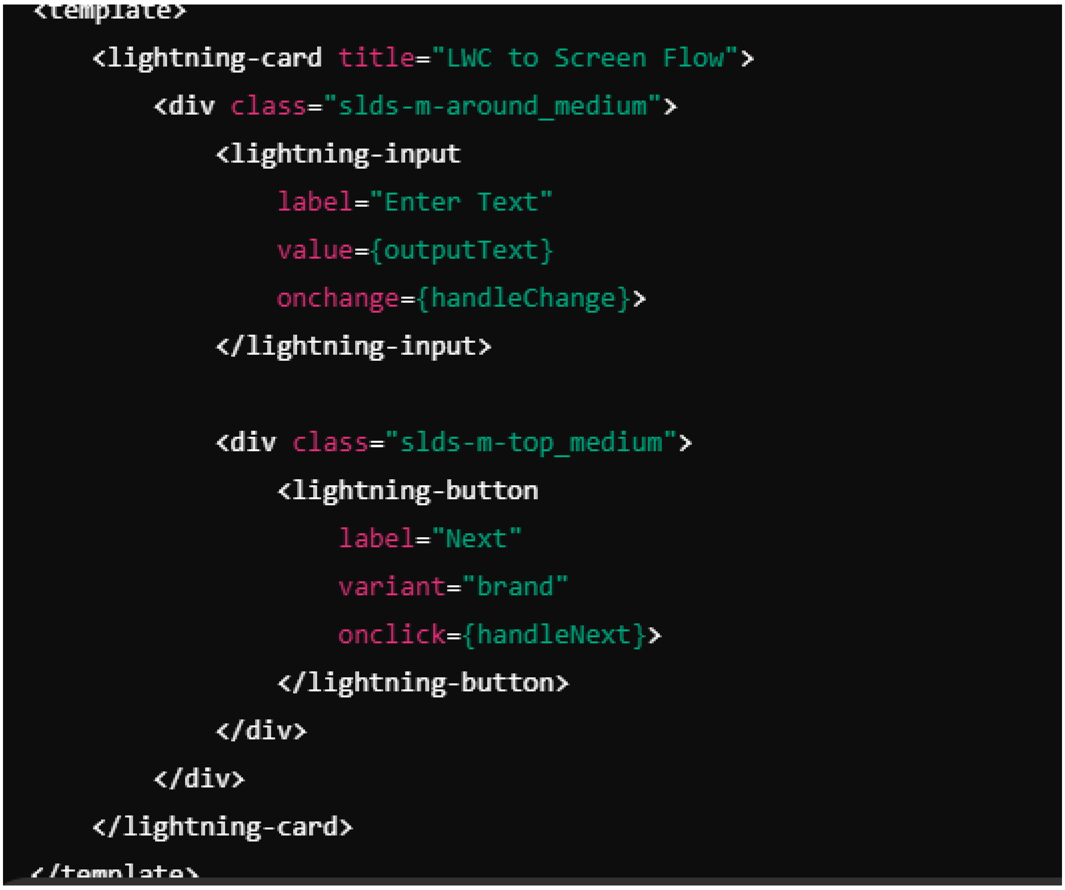
To use an LWC inside a Screen Flow, the LWC component must be designed as a **Flow Screen Component**. This requires implementing the `lightning-flow-supports` interface.

Steps to Implement LWC in Screen Flow

1. **Create a Lightning Web Component**
   - Implement the `lightning-flow-supports` interface to make the LWC available in Flow.
   - Define properties to send and receive data from Flow.
2. **Expose the LWC as a Flow Component**
   - Use `@api` decorators for input and output properties.
   - Ensure proper event handling for communication.
3. **Use the LWC in a Screen Flow**
   - Add the component inside a Screen Flow.
   - Map input and output variables to Flow elements.

Example LWC

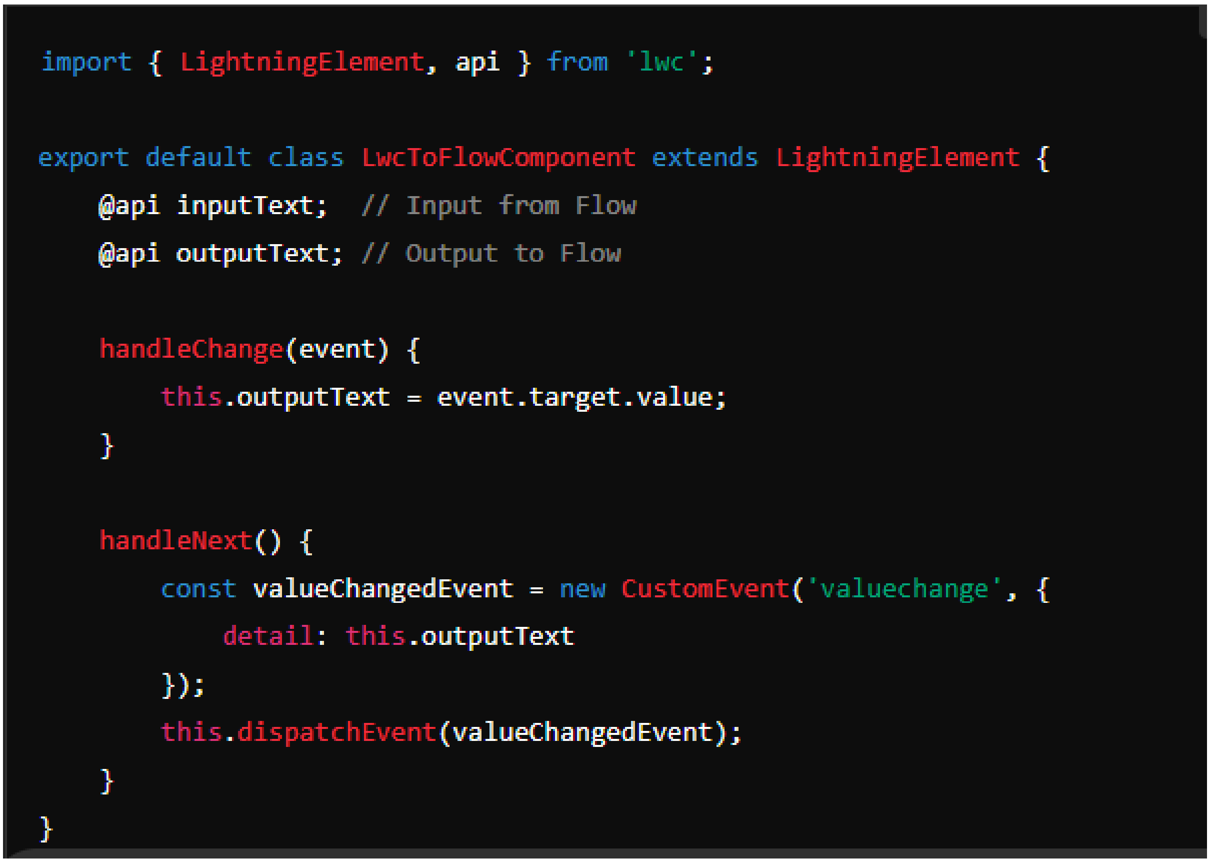
HTML



```
<template>
    <lightning-card title="LWC to Screen Flow">
        <div class="slds-m-around_medium">
            <lightning-input
                label="Enter Text"
                value={outputText}
                onchange={handleChange}>
            </lightning-input>

            <div class="slds-m-top_medium">
                <lightning-button
                    label="Next"
                    variant="brand"
                    onclick={handleNext}>
                </lightning-button>
            </div>
        </div>
    </lightning-card>
</template>
```

JAVASCRIPT



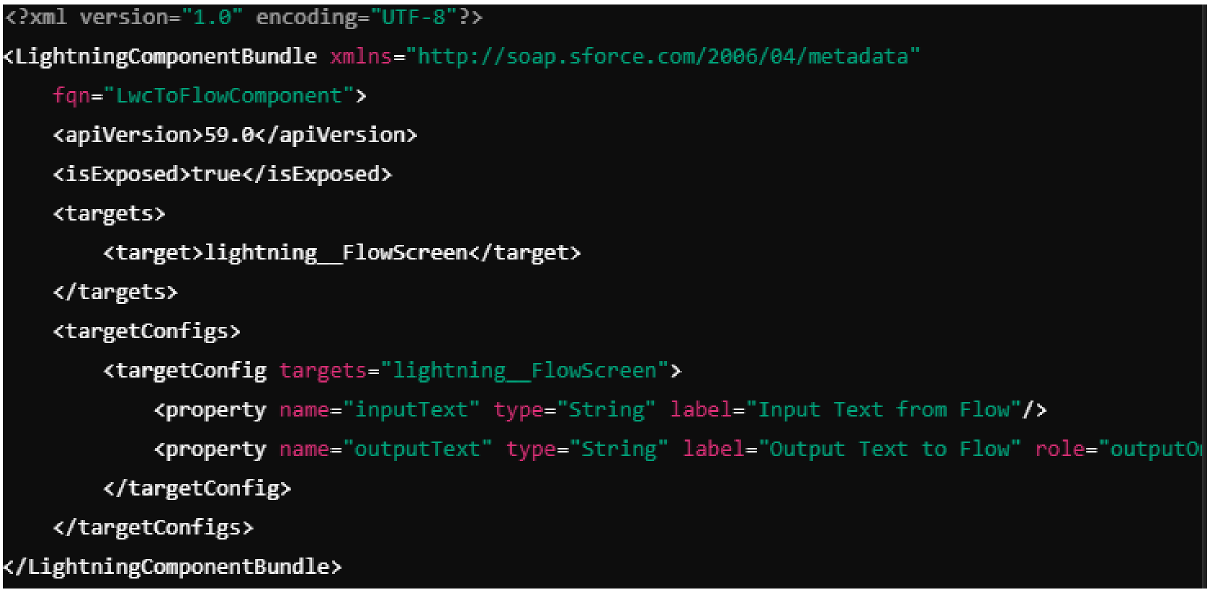
```javascript
import { LightningElement, api } from 'lwc';

export default class LwcToFlowComponent extends LightningElement {
    @api inputText;  // Input from Flow
    @api outputText; // Output to Flow

    handleChange(event) {
        this.outputText = event.target.value;
    }

    handleNext() {
        const valueChangedEvent = new CustomEvent('valuechange', {
            detail: this.outputText
        });
        this.dispatchEvent(valueChangedEvent);
    }
}
```

XML



```xml
<?xml version="1.0" encoding="UTF-8"?>
<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata"
    fqn="LwcToFlowComponent">
    <apiVersion>59.0</apiVersion>
    <isExposed>true</isExposed>
    <targets>
        <target>lightning__FlowScreen</target>
    </targets>
    <targetConfigs>
        <targetConfig targets="lightning__FlowScreen">
            <property name="inputText" type="String" label="Input Text from Flow"/>
            <property name="outputText" type="String" label="Output Text to Flow" role="outputO
        </targetConfig>
    </targetConfigs>
</LightningComponentBundle>
```

Adding LWC to Screen Flow

1. Navigate to **Salesforce Flow Builder**.
2. Create a **Screen Flow**.

3. Add a **Screen Element**.
4. Drag and drop the **LWC Component** into the screen.
5. Map **Flow Variables** to LWC **input and output properties**.
6. Save and activate the Flow.

Where Can We Call LWC in Salesforce?

Use Case	Can We Call LWC?	How to Implement?
Lightning Record Page	**Yes**	Add LWC to **Lightning App Builder** and place it on a record page.
Lightning Home Page	**Yes**	Add LWC to **Lightning App Builder** and use on home page.
Lightning App Page	**Yes**	Use LWC in a **Lightning App Page** for custom UI.
Screen Flow (Flow Builder)	**Yes**	Expose LWC to **Flow Screen** using `<target>lightning__FlowScreen</target>` in `meta.xml`.
Record-Triggered Flow	**No** (Directly)	Use **Apex Invocable Method** or **Custom Notification** to trigger LWC indirectly.
Auto-Launched Flow	**No** (Directly)	Use **Apex + Platform Events** to notify LWC.
Scheduled Flow	**No** (Directly)	No UI support, but can trigger **Custom Events**.
Lightning Quick Action	**Yes**	Create **LWC Quick Action** (`lightning__RecordAction`) and place it on an object record page.
Lightning Utility Bar	**Yes**	Add LWC to the **Utility Bar** for always-accessible UI.
Salesforce Tab (Custom Tab)	**Yes**	Expose LWC as a **Custom Tab** and add to Navigation Menu.
Experience Cloud (Community)	**Yes**	Use LWC in **Experience Builder** to display on Community pages.
Apex AuraEnabled Call	**Yes**	Call Apex using `@AuraEnabled` method inside LWC.
Apex Imperative Call	**Yes**	Use `@wire` or `fetch()` to get Apex data in LWC.
Visualforce Page	**Yes**	Use LWC inside **Lightning Out** in Visualforce.

Use Case	**Can We Call LWC?**	**How to Implement?**
Lightning Message Service (LMS)	**Yes**	Use LMS to **communicate between LWC components** or across pages.
Navigation to LWC via URL	**Yes**	Use **Custom Button + URL Parameters** to open LWC with data.
Embedded in Aura Component	**Yes**	Use `<c-myLwcComponent></c-myLwcComponent>` inside **Aura Component**.
Embedded in VF Page (via Aura)	**Yes**	Use **Lightning Out** to embed LWC in **Visualforce Page**.
OmniStudio (Vlocity)	**Yes**	Use LWC as a **FlexCard or OmniScript Component**.

- **LWC works directly** in most UI-based features like **Lightning Pages, Screen Flows, Quick Actions, Tabs, and Experience Cloud.**

- **LWC does NOT work directly** in **Record-Triggered Flows, Auto-Launched Flows, or Scheduled Flows**, but can be triggered **indirectly** via **Apex, Custom Notifications, or Platform Events.**

- **LWC is highly flexible** and can be embedded in **Aura Components, Visualforce Pages, OmniStudio, and more.**