**Parent to Child Component Communication in LWC**

Lightning Web Components (LWC) is a modern, standards-based framework for building web applications on the Salesforce platform. LWC promotes the development of modular, reusable components that can be easily integrated into your applications. One of the key aspects of component-based development is communication between components. In this blog, we will explore how to achieve Parent to Child component communication in LWC.

* **Component Composition**:   
  Component composition in Lightning Web Components (LWC) is a fundamental concept that involves creating a hierarchy of components where one component can include or "compose" other components. This allows you to build complex and reusable user interfaces by breaking them down into smaller, more manageable parts.  
    
  **Parent Component**: This is the higher-level component that includes and manages one or more child components. The parent component can pass data to its child components, listen for events from them, and control their behaviour.

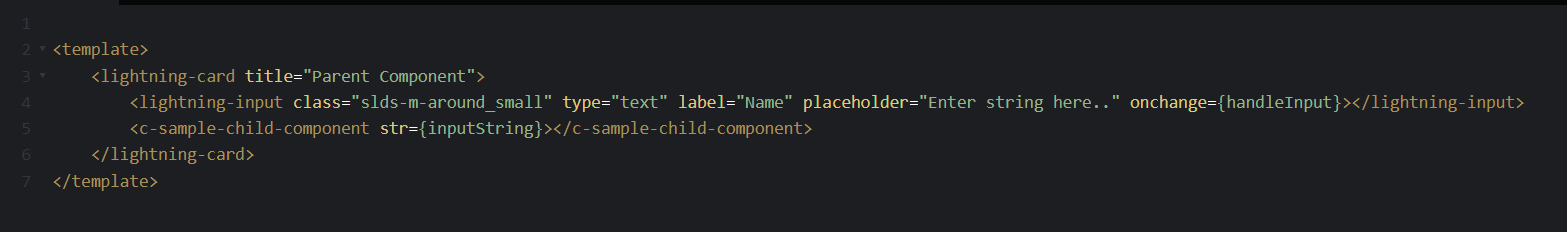
**Child Components**: These are the smaller, self-contained components that can be included within the parent component. Child components receive data from the parent component, handle their own logic, and emit events that the parent component can listen to.

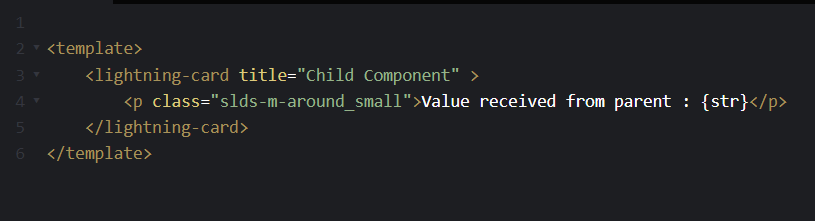
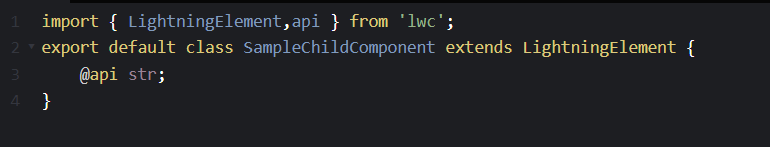
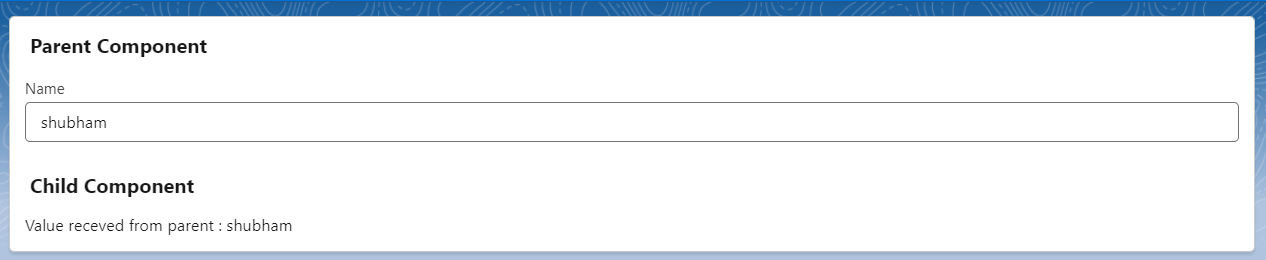
* **Including Child Components in a Parent Component**:  
  Syntax to embed child component into parent component is as follows :  
  Parent.html  
  <template>  
   <c-child-component></c-child-component>  
  </template>
* **Why Component Communication is Essential**

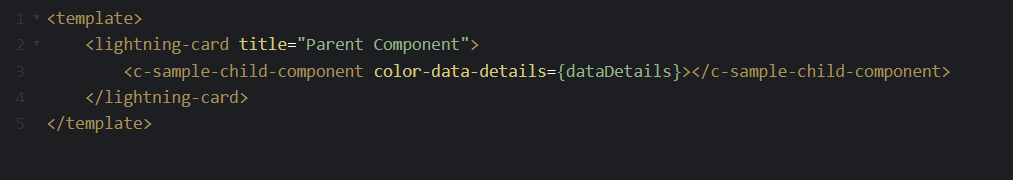
In a complex web application, it's common to break the user interface into smaller, manageable components. These components need to work together to provide a seamless user experience**.**

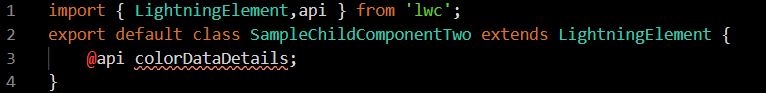
Parent to Child component communication is particularly important when you want to pass data, trigger actions, or share information between a parent component and its child components.

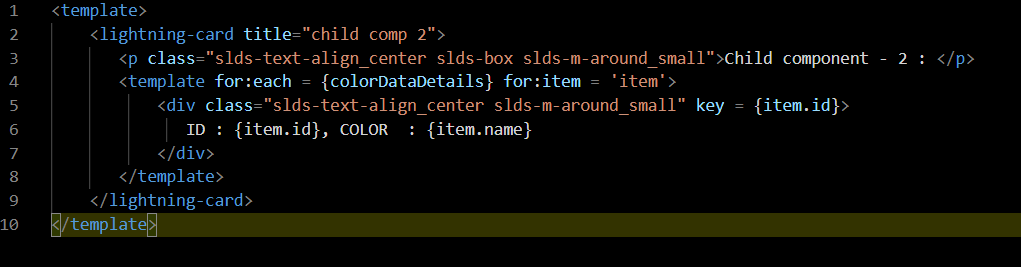
* **Passing Data from Parent to Child**

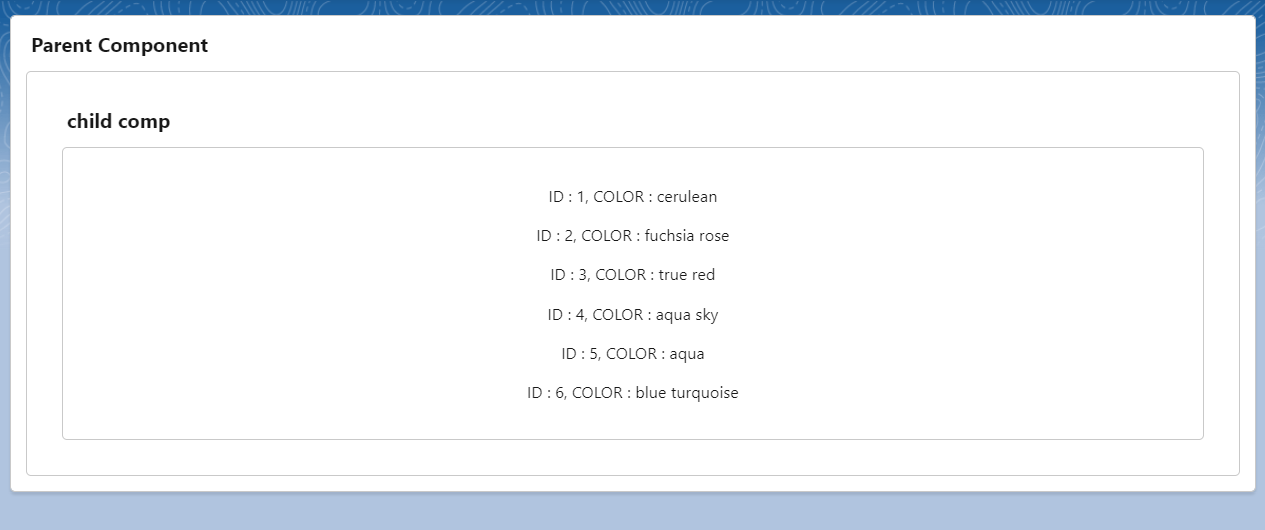
There are several methods for passing data from a parent component to a child component in LWC**.** The most commonly used method is by using attributes. In order to implement this communication, properties in the child component need to be accessible at parent component. To expose properties at child component publicly, **@api** decorator is used. Using API decorator primitive as well as non-primitive data at child component gets accessible to parent component.  
  
USE CASE I: Parent to child communication using primitive data:   
sampleParentComponent.html  
sampleParentComponent.js  


sampleChildComponent.html  
sampleChildComponent.js  
  
in this example, we are taking input using lightning input in parent component and passing that input to child component. As you can see, @api decorator is used to make the variable in child component accessible to parent component.  
  
OUTPUT:  


USE CASE II: Parent to child communication using non-primitive data:   
sampleParentComponent.html  
  
sampleParentComponent.js  


sampleChildComponent.js  


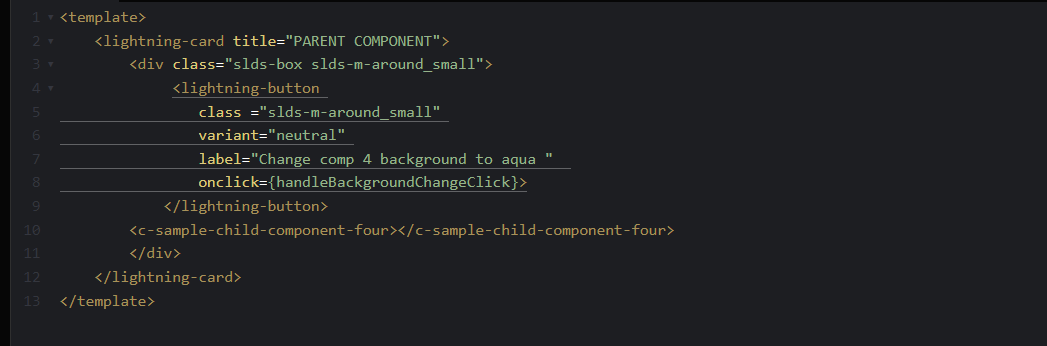
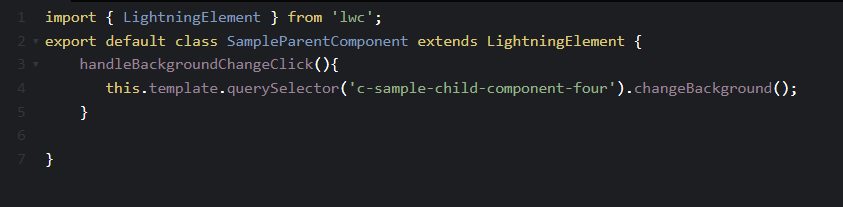
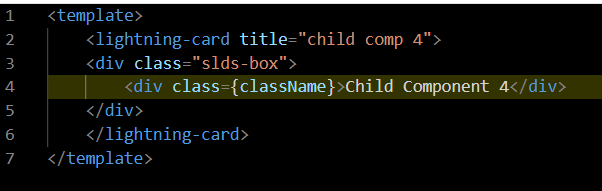
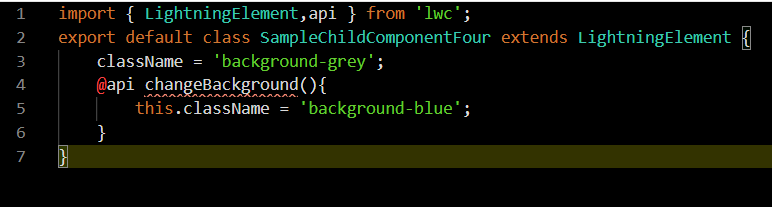
sampleChildComponent.html  


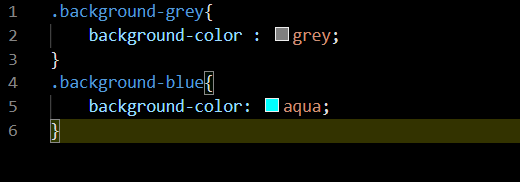
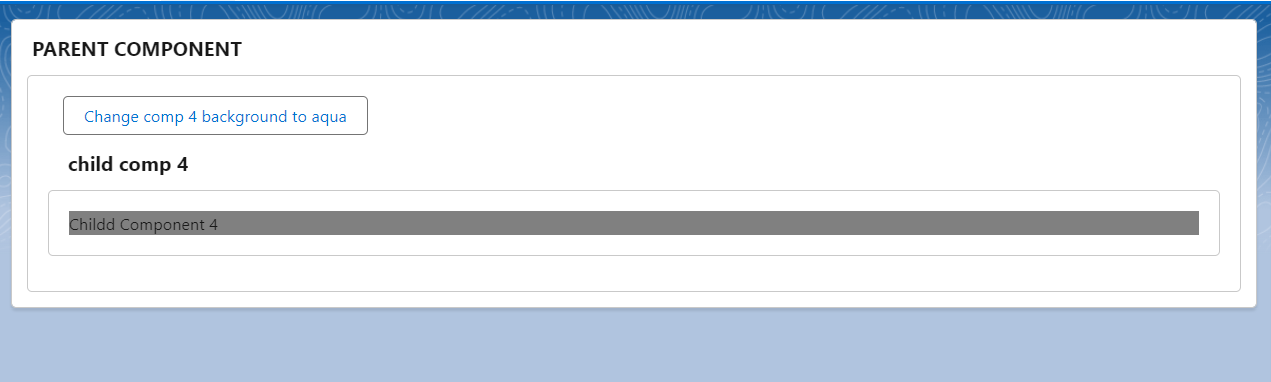
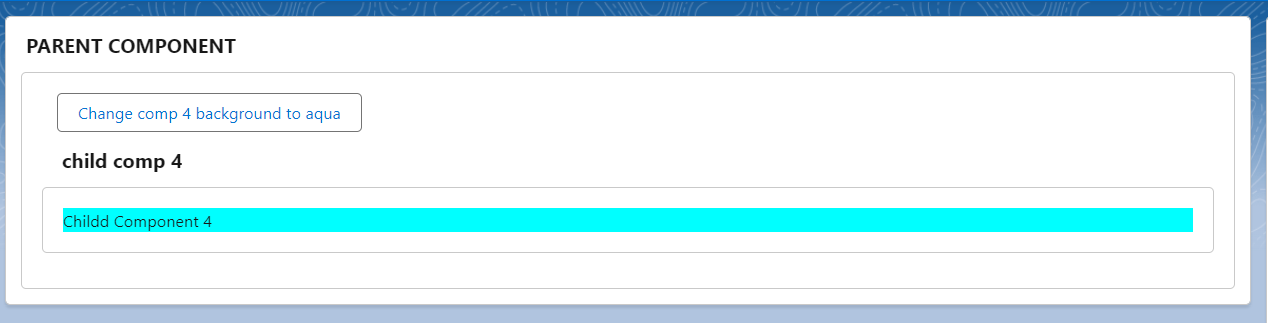
OUTPUT :  


In this example , we have passed a list From parent component to child component . As you can see the property in child component is decorated using @api. While passing values from parent to child , parameter needs to be mentioned in kebab case.

USE CASE III: Parent to child communication by calling child component methods from Parent

This can be achieved using **template.Queryselector(‘child-component’).methodName();**In this case, the method which is being called in parent component needs to be decorated using @api.

sampleParentComponent.html sampleParentComponent.js   
  
sampleChildComponentFour.html  
   
sampleChildComponentFour.js  


sampleChildComponentFour.css  
  
OUTPUT:  
  


* **Conclusion**

Effective component communication is essential for building modular and maintainable applications in LWC. By using attributes, properties, custom events, and shared JavaScript modules, you can establish robust Parent to Child communication. Understanding these techniques will empower you to create more interactive and cohesive Lightning Web Components for your Salesforce applications.