**Lightning Component Attribute** (aura:attribute) are like member variables on a class in Apex. `<aura:attribute>` tag is used to add an attribute to the [lightning component](https://salesforce.com) and Salesforce lightning App.

- Each attribute must have "name" and "type".
- To make attribute as required use required="true" in the tag.
- To make default value use default="string or integer".

**Attribute naming Rules.**

An attribute name must follow these naming rules.

- Must begin with a letter or an underscore.
- Must contain only alphanumeric or underscore characters.

---

In this Salesforce tutorial, we are going to learn about Lightning Component attributes and how to use `<aura:attribute>` tag in Salesforce lightning app and lightning component.

In this Salesforce example, we are going to create an application and a component called “addition”. Here we will add two integers and will find the sum of two integers. And also we add styling to the Salesforce lightning component.

Navigate to Developer console and create new Lightning component.

Here we have to define two tags they are `<aura:component>` and `<aura:attribute>`. This lightning Component Attribute will have name and type.

```html
<aura:component>
  <h1>Addition of Two Numbers</h1>
  <aura:attribute name="num1" type="integer" default="100" />
  <aura:attribute name="num2" type="integer" default="20" />
  <aura:attribute name="sum" type="integer" />
  <br/>
  <div class="addC">
    {{v.num1}} + {{v.num2}} = {{v.sum}}
  </div>
  <br/>
  <p>
    <ui:button label="submit" press="{!c.add}" />
  </p>
</aura:component>
```
As shown above, we have selected type as integer because we are storing numbers. And if we want to provide default values enter default="value". To display two numbers we have to use particular expression like \{!v.num1\} + \{!v.num2\} = \{!v.sum\}.

- We also defined a button called submit. When a submit button is clicked the result will be displayed on the screen.
- The sum value will not be displayed in the output when we click on submit because we have not defined what should happen when submit button is clicked.

Here we have to add a call a method called \{!c.add\}. Now click on controller in the component and add the controller as shown below.

When we click on Controller, Controller file will be created automatically. We want to call \{!c.add\} function.
Click on **Save** button.

lightning **Component** Attribute output.

When we preview the lightning application, the output will be as follow.

![Addition of Two Numbers](https://tutorialskart-dev-ed.lightning.force.com/tutorialskart/lightning.app)

The look and feel of the output will be changes by adding style to lightning component. Click on Style and add the following css code given below.

```javascript
({
    add: function(component) {
        var addval=component.get("v.num1")|component.get("v.num2");
        component.set("v.sum",addval);
    }
})
```

```
.THIS {
    background-color: white;
    font-family:sans-serif;
    text-align:center;
}

body.THIS {
    border: 2px solid grey;
}

H1.THIS {
    background-color: white;
    font-weight:strong;
    width: 20%;
    border: 1px solid green;
    padding: 10px;
    font-family:sans-serif;
    text-align:center;
    margin:auto;display:block;
}

body.THIS {
    text-align:center;
}

p.THIS {
    text-align: center;
}

button.THIS {
    position: absolute;
    top: 50%;
}

addC.THIS {
    margin:auto;
    display:block;
}
```
Conclusion: In this Salesforce tutorial, we have learned about lightning Component Attribute and created Salesforce lightning component for addition of two integer. In our upcoming Salesforce training, we will learn about Salesforce lightning Component composition with an example.
<table>
<thead>
<tr>
<th>Section 5: Setting up User Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page Types and Page Elements</td>
</tr>
<tr>
<td>What are page Layouts?</td>
</tr>
<tr>
<td>Home page layouts and components</td>
</tr>
<tr>
<td>Why Documents are used in Salesforce</td>
</tr>
<tr>
<td>List Views</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Section 6: Customizing Standard Salesforce Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>How to Customise Standard Salesforce application</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Section 7: Salesforce Security Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview on Salesforce security Model</td>
</tr>
<tr>
<td>How to create New Users in SFDC?</td>
</tr>
<tr>
<td>What is SAML?</td>
</tr>
<tr>
<td>Public groups in Salesforce</td>
</tr>
<tr>
<td>Role Hierarchies in Salesforce.com</td>
</tr>
<tr>
<td>How to create and manage Profiles</td>
</tr>
<tr>
<td>What is Field Level Security?</td>
</tr>
<tr>
<td>How to use Permission Sets?</td>
</tr>
<tr>
<td>Control Record visibility using Organisation wide Defaults</td>
</tr>
<tr>
<td>Control Record visibility using Role Hierarchy</td>
</tr>
<tr>
<td>Control Record visibility using Sharing Rules</td>
</tr>
<tr>
<td>What are Record Types and how can we configure</td>
</tr>
<tr>
<td>What are Page Types and page Elements</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Section 8: Data Management</th>
</tr>
</thead>
<tbody>
<tr>
<td>Different Data Management Tools</td>
</tr>
<tr>
<td>How to install Apex Data Loader in macOS and Windows</td>
</tr>
<tr>
<td>How to Insert a record using Apex Data Loader</td>
</tr>
<tr>
<td>How to Update and Insert records Apex Data Loader</td>
</tr>
<tr>
<td>How to Delete and Export records Apex Data Loader</td>
</tr>
</tbody>
</table>
Creating first Salesforce lightning App
Styling Salesforce lightning App
Lightning Component: aura:attribute tag
What is Salesforce DX?
Create Salesforce DX project
Salesforce Dev Hub Setup step-by-step
Creating Scratch Org