

Aura Component Syntax Cheat Sheet

Component Basics

```
<aura:component implements="force:appHostable,flexipage:availableForAllPageTypes">
    <!-- your code -->
</aura:component>

<aura:attribute name="someValue" type="String" default="Hello" />
<aura:attribute name="isVisible" type="Boolean" default="true" />
<aura:attribute name="items" type="List" />
<aura:attribute name="record" type="Object" />
```

Event Declaration

```
<!-- Component Event -->
<aura:event type="COMPONENT">
    <aura:attribute name="message" type="String" />
</aura:event>

<!-- Application Event -->
<aura:event type="APPLICATION">
    <aura:attribute name="message" type="String" />
</aura:event>
```

Component Event Usage

```
<!-- Register in child -->
<aura:registerEvent name="compEvent1" type="c:compEvent" />

<!-- Fire in controller -->
var evt = component.getEvent("compEvent1");
evt.setParams({ message: "Hello from child" });
evt.fire();

<!-- Handle in parent -->
<aura:handler name="compEvent1" event="c:compEvent" action="{!c.handleComponentEvent}" />
```

Application Event Usage

```
<!-- Fire -->
var appEvt = $A.get("e.c:applicationEvent");
appEvt.setParams({ message: "Broadcast msg" });
appEvt.fire();

<!-- Handle -->
<aura:handler event="c:applicationEvent" action="{!c.handleApplicationEvent}" />
```

Common Expressions

```
{!v.attributeName}    <!-- Access attribute -->
{!c.controllerMethod}  <!-- Controller method -->
```

Lightning UI Snippets

Aura Component Syntax Cheat Sheet

```
<lightning:button label="Click Me" onclick="{!c.handleClick}" />
<lightning:card title="My Card">
    <p class="slds-p-horizontal_small">Hello World</p>
</lightning:card>
<lightning:input label="Name" value="{!v.name}" />
```

JS Controller & Helper

```
({
  doInit : function(component, event, helper) {},
  handleClick : function(component, event, helper) {}
})
```

Helper:

```
({
  helperMethod : function(component, event) {}
})
```

Renderer

```
({
  afterRender: function (component, helper) {
    this.superAfterRender();
    // custom logic
  }
})
```

Conditions & Loops

```
<aura:if isTrue="{!v.isVisible}">
    <p>Visible</p>
<aura:set attribute="else">
    <p>Hidden</p>
</aura:set>
</aura:if>
```

```
<aura:iteration items="{!v.items}" var="item">
    <p>{!item.name}</p>
</aura:iteration>
```

Apex Integration

```
// Apex Class
@AuraEnabled public static String getMessage() {
    return 'Hello from server!';
}

// JS Call
var action = component.get("c.getMessage");
action.setCallback(this, function(response) {
    if (response.getState() === "SUCCESS") {
        component.set("v.message", response.getReturnValue());
    }
});
```

Aura Component Syntax Cheat Sheet

```
    }  
  });  
  $A.enqueueAction(action);
```