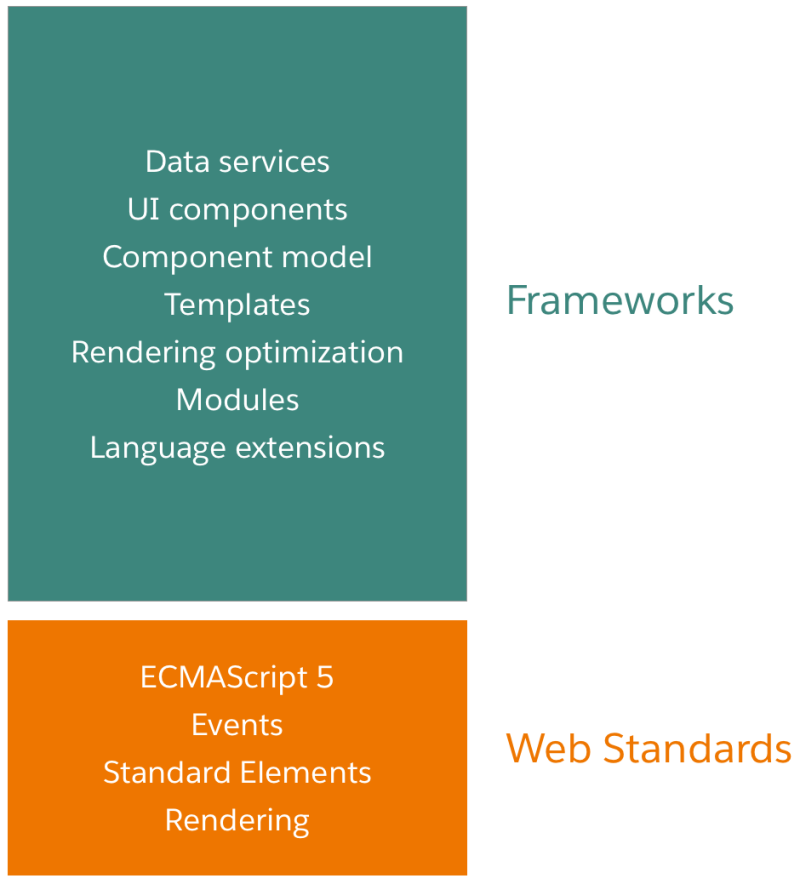
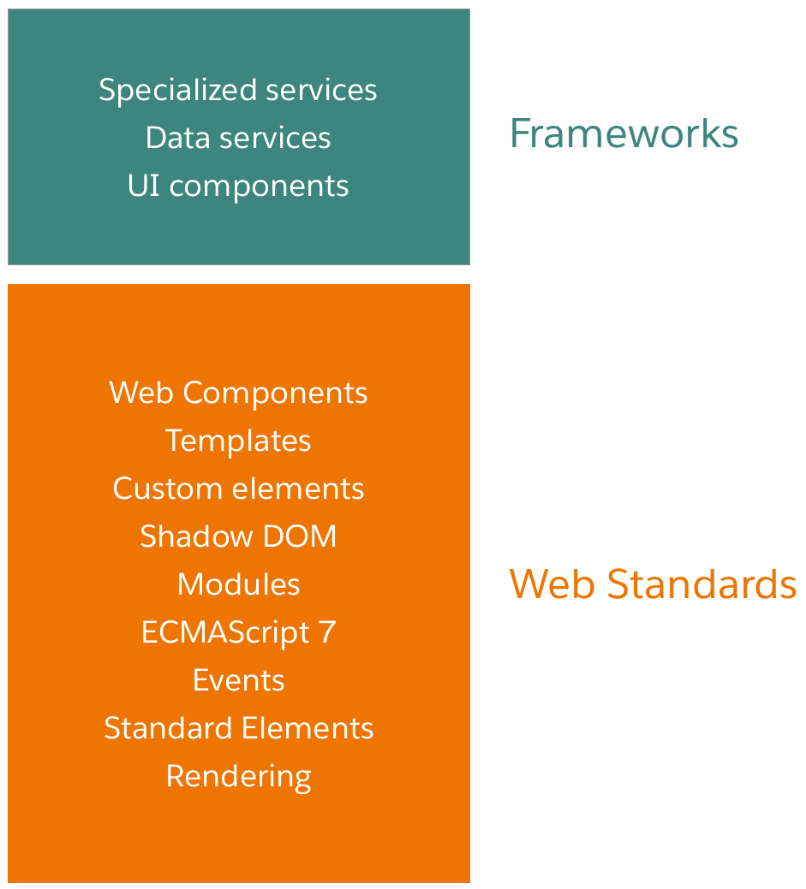
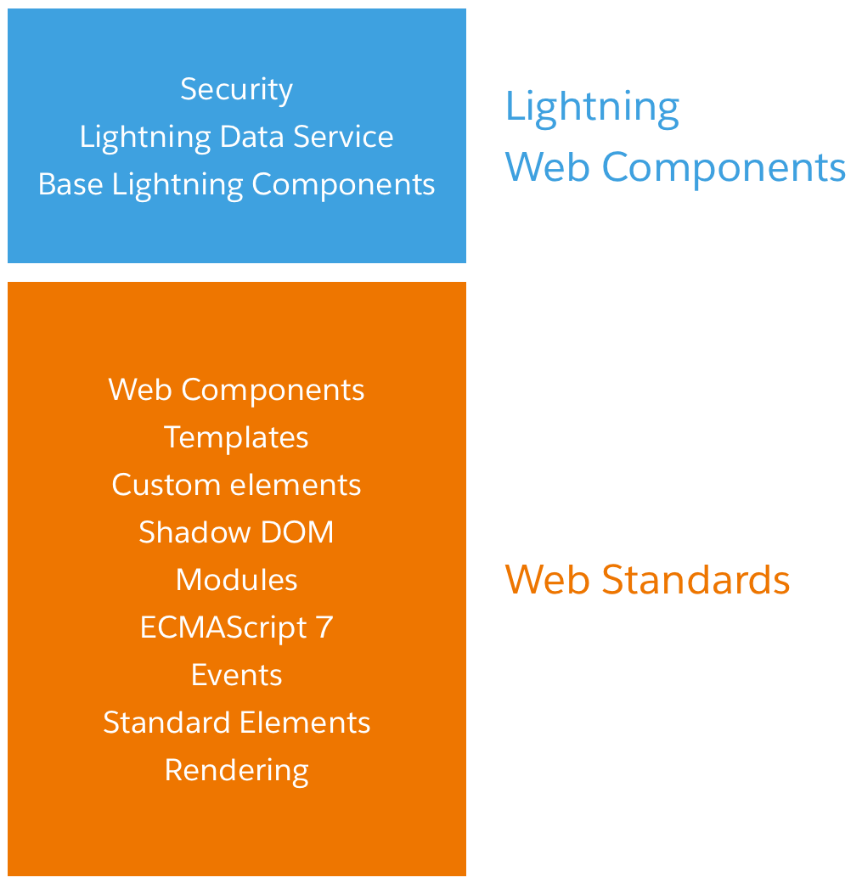
Web standards in 2014

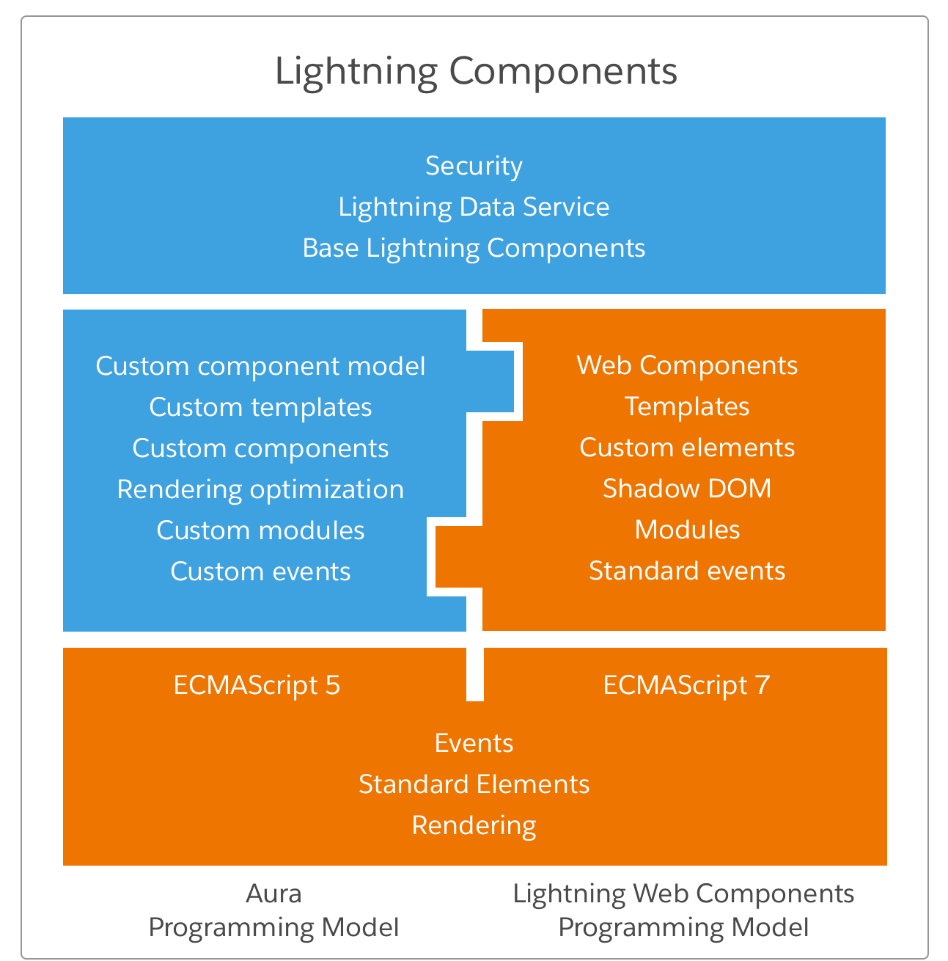


Web standards in 2019



LWC





Camel => gyanEasy  
Kebab => gyan-easy  
Pascal => GyanEasy

Camel => amitLuniya  
Kebab => amit-luniya  
Pascal => AmitLuniya

Component folder

myComponent

├──myComponent.html

├──myComponent.js

├──myComponent.js-meta.xml

├──myComponent.css

└──myComponent.svg

Let’s say there is component called myComponent and want to call from another component.. HTML file

<template>

<c-my-component></c-my-component>

</template>

Javascript file name myComponent => MyComponent.js   
import and export

import { LightningElement } from 'lwc';

export default class MyComponent extends LightningElement {

}

Config file => myComponent.js-meta.xml

*<?xml version="1.0" encoding="UTF-8"?>*

<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">

<apiVersion>45.0</apiVersion>

<isExposed>false</isExposed>

</LightningComponentBundle>

CSS file => myComponent.css

SVG file => myComponent.svg

Additional javascript file.. you can name anything..

Test files..

myComponent

|--myComponent.html

|--myComponent.js

|--myComponent.js-meta.xml

|--myComponent.css

|--\_\_tests\_\_

|--myComponent.test.js

Decorators –   
@track - it makes property private and reactive  
@api - it marks a property public and reactive  
@wire - To read Salesforce data, LWC uses a reactive wire service.

Data binding in aura components have bound and unbound expressions but in LWC it is one way communication. To communicate down from a parent component to a child component, set a property or call a method on the child component. To communicate up from a child component to a parent component, send an event.

JS config file tags and targets,

Each Lightning web component folder must include a configuration file named *<component>*.js-meta.xml. The configuration file defines the metadata values for the component, including the design configuration for the Lightning App Builder and Community Builder.

**apiVersion**

A double value that binds the component to a Salesforce API version.

**description**

A short description of the component, usually a single sentence. Appears in list views, like the list of Lightning Components in Setup, and as a tooltip in the Lightning App Builder and in Community Builder.

**isExposed**

If isExposed is false, the component is not exposed to Lightning App Builder or Community Builder.

To allow the component to be used in Lightning App Builder or Community Builder, set isExposed to true and define at least one <target>, which is a type of Lightning page.

**masterLabel**

The title of the component. Appears in list views, like the list of Lightning Components in Setup, and in the Lightning App Builder and in Community Builder.

**targets**

Specifies where the component can be added, such as on a type of Lightning Page or in Embedded Service Chat. If you want your component to appear in the Lightning App Builder or in Community Builder, specify at least one Lightning page type.

Supports the target subtag.

**target**

A Lightning page type. Valid values are:

| **VALUE** | **DESCRIPTION** |
| --- | --- |
| lightning\_\_AppPage | Enables a component to be used on an App page in Lightning App Builder. |
| lightning\_\_HomePage | Enables a component to be used on a Home page in Lightning App Builder. |
| lightning\_\_RecordPage | Enables a component to be used on a record page in Lightning App Builder. |
| lightning\_\_UtilityBar | Enables a component to be used as a utility item on the utility bar in the App Manager. |
| lightning\_\_FlowScreen | Enables a component to be used on flow screens in Flow Builder. |
| lightning\_\_Tab | Enables a component to be used in a custom tab in Lightning Experience or the Salesforce mobile app. |
| lightning\_\_Inbox | Enables a component to be used in Lightning App Builder to add to email application panes for the Outlook and Gmail integrations. |
| lightningCommunity\_\_Page | Enables a component to be used on a Lightning community page in Community Builder. |
| lightningCommunity\_\_Default | Used together with lightningCommunity\_\_Page. Enables a component that includes configurable properties to be used on a Lightning community page in Community Builder. When the component is selected on the page, the properties appear. |
| lightningSnapin\_\_ChatMessage | Enables a component to be selected from Embedded Service Chat Setup. If a component imports the lightningsnapin/baseChatMessage module, it must specify the lightningSnapin\_\_ChatMessage target. For more information, see the [lightningsnapin/baseChatMessage](https://developer.salesforce.com/docs/component-library/bundle/lightningsnapin-base-chat-message/documentation" \t "_blank) documentation. |

**targetConfigs**

Configure the component for different page types and define component properties. For example, a component could have different properties on a record home page than on the Salesforce Home page or on an app page. For Communities, only lightningCommunity\_\_Default supports component properties.

Supports the targetConfig subtag.

**targetConfig**

Use a separate targetConfig for each different page type configuration. Specify one or more page types in the targets attribute, such as <targetConfig targets="lightning\_\_RecordPage"> or <targetConfig

targets="lightning\_\_RecordPage,lightning\_\_AppPage">.

The targets attribute value that you specify must match one or more of the page types that you listed under <targets> for the component.

Supports the property, objects, and supportedFormFactors subtags.

**property**

Specifies a public property of a component that can be set in Lightning App Builder, App Manager, Lightning Flow Builder, or Community Builder. The component author defines the property in the component’s JavaScript class using the @api decorator.

The property tag supports these attributes:

| **ATTRIBUTE** | **DESCRIPTION** | **REQUIRED** |
| --- | --- | --- |
| color | Displays a color picker. Specify RGBA, RBB, or hex strings. For example, rgba(0, 255, 255,  1).  <property name="backgroundColor" type="Color" default="rgba(0, 255, 255, 1)"> | No |
| datasource | Renders a field as a picklist, with static values. Supported only if the type attribute is String.  <property name="Name" datasource="value1,value2,value3" />  You can also set the picklist values dynamically using an Apex class.  <property name="Name" datasource="apex://MyCustomPickList"/>  See [Create Dynamic Picklists for Your Custom Components](https://developer.salesforce.com/docs/atlas.en-us.lightning.meta/lightning/components_config_for_app_builder_dynamic_picklists.htm) for more information. | No |
| default | The default value for the attribute. | No |
| description | Displays as an i-bubble for the attribute in the tool. | No |
| label | Displays as a label for the attribute in the tool. | No |
| max | The maximum allowed value for an attribute of type Integer. Not supported if the target is lightning\_\_FlowScreen. | No |
| min | The minimum allowed value for an attribute of type Integer. Not supported if the target is lightning\_\_FlowScreen. | No |
| name | Required if you’re setting properties for your component. The attribute name. This value must match the property name in the component’s JavaScript class. | Yes |
| placeholder | Input placeholder text for the attribute when it displays in the tool. This text is the ghost text in text fields and text areas before a user starts typing. Supported only if the type attribute is String. | No |
| required | Specifies whether the attribute is required. The default value is false. | No |
| role | Specifies whether the attribute is inputOnly or outputOnly. If you don’t specify the role attribute, the default value allows input and output. For example, if a property is restricted to outputOnly, users can’t set its value from a Lightning record page. Supported only if the target is lightning\_\_FlowScreen. | No |
| type | The attribute’s data type.  To specify a list, add array notation [] to the end of any data type.  These values are valid for all targets.   * Boolean * Integer * String   These values are valid only if the target is lightning\_\_FlowScreen.   * apex://*namespace*.*Classname*—An Apex class used for an Apex-defined data type. * Date—A date in ISO 8601 format. * DateTime—A datetime in ISO 8601 format. * @salesforce/schema/*namespace*.*Objectname*—An object.   *Classname*—The name of the Apex class.  *namespace*—The namespace of the Salesforce organization. Specify a namespace unless the organization uses the default namespace (c), in which case don’t specify it.  *Objectname*—The name of the object. | Yes |

**objects**

A set of one or more objects the component is supported for. This tag set works only inside a parent targetConfig that’s configured for lightning\_\_RecordPage. Specify the objects tag set only once inside a targetConfig set. Supports the object subtag.

**object**

Defines which objects the component is supported for. Use one object tag for each supported object. You can’t use ‘\*’ to denote all objects.

See the [User Interface API Developer Guide](https://developer.salesforce.com/docs/atlas.en-us.uiapi.meta/uiapi/ui_api_get_started_supported_objects.htm) for the list of supported objects.

**supportedFormFactors**

A set of one or more form factors or devices that the component supports. Specify the supportedFormFactors tag set once inside a targetConfig set. Supports the supportedFormFactor subtag.

**supportedFormFactor**

Defines which form factor or device the component supports. Use one supportedFormFactor tag for each supported form factor.

The supportedFormFactor tag supports these attributes:

| **ATTRIBUTE** | **DESCRIPTION** | **REQUIRED** |
| --- | --- | --- |
| type | Device form factor on which the component displays. Valid values are:   * Large—Represents the desktop form factor. Supported for all page types. * Small—Represents the phone form factor. Supported for lightning\_\_AppPage only. |  |