
OurUniversity
EFMD CLIP ACCREDITED
Force.com Visualforce Pages
salesforce
global strategic
consulting partner
Lesson 04 : Form and Output Components
People matter, results count.

Lesson Objectives

By the end of this lesson, you will be able to :

- List Key tags and define what their attributes control
- Create Visualforce pages that use these tags to create page output tables and lists

Form Components
<apex:form>

This tag enable a section of a Visualforce page to allow users to enter data submit it with commandButton or commandLink

Attribute include:

- accept (String): a comma-separated list of content types that this form can handle; values can include text/html, image/png, video/mpeg, text/css, audio/basic, and more.
- title (String): the text to display as a tooltip when the user's mouse pointer hovers over this component
- target (String): name of the frame that displays the response after the form is submitted; possible values include _blank, _parent, _self, and _top

Form Components
<apex:inputField>

This tag corresponds to a Salesforce object field and respects the attributes of that field.

- The associated UI widget is automatically used (calenders for date fields, etc)
- The outputField version makes the field values read-only

When used within a pageBlockSection, the label is automatically displayed as well.

Attribute include:

- required (Boolean): specifies if field is required on this page (false)
- value (Object): merge field that references Salesforce field in the form of (!object.field)

Form Components <apex:inputWidget>

These tags provide input widgets for data that does not correspond to a Salesforce object field to be used with pageBlockSectionItem tags.

These include:

- <apex:inputCheckbox>
- <apex:inputHidden> (useful for passing variables between pages)
- <apex:inputSecret> (for masked data, such as passwords)
- <apex:inputText>
- <apex:inputTextarea> (includes an attribute to specify saving the text as plain text or rich text)

Form Components
<apex:inputWidget>

In general, these tags include attributes that give you ways to:
Vary the size of the data and the widgets
Make the inputs required
Control the tabbing sequence and keyboard shortcuts
Bind the data to custom controllers
Handle standard JavaScript events

Form Components
<apex:selectWidget> Components

These are a series of additional tags to support the display of UI widget in organized tables.

These include:

- <apex:selectCheckboxes>
- <apex:selectList>
- <apex:selectRadio>

These use <apex:selectOption> and <apex:selectOptions> to create option lists for the other selection tags.

These work with the Apex SelectOption object to specify the values in custom controllers.

Form Components
<apex:inputFile>

This tag allows users to upload files and turn them into:

- Attachments on records
- Documents
- Apex Blobs

Attributes include:

- accept (String): a comma-separated list of file types accepted for upload
- contentType (String): the uploaded file's content type
- fileName (String): the name of the uploaded file

Form Components
<apex:commandButton> & <apex:commandLink>

These tags create buttons and links that allow the users to navigate and take actions.

- These tags must be used within a form tag.

Attributes include:

- action (Action Method): references through merge-field syntax a method on the controller. Standard controllers support {!save}, {!edit}, {!delete}
- Buttons for standard controllers are not rendered for users that do not have appropriate permissions
- Value (String): text displayed on the button or link

The <apex:param> tag can be used to dynamically add URL parameters to link

Output, Table and List Components
Output Components

These tags all deal with displaying information without allowing the user to change any data.

Many output components have parallel input components such as **inputField** and **outputField.**

Output, Table and List Components <apex:outputLabel>

This tag creates a label for input or output widgets that do not automatically come with a label.

- This is usually because the data is not coming from Salesforce or a non-default UI widget is being used for a Salesforce field (i.e., not outputField)

Atrributes include:

- value (String): the text displayed as the label
- for (String): Id of the component with which the label should be associated
- tabindex (Integer): the order in which this label/field comes into focus using the **Tab** key

Demo

Overriding a default Edit Page with alternative UI widgets

Output, Table and List Components <apex:outputField>

This tag creates a read-only display of a label and value for a Salesforce field.

The data is automatically formatted according to the field type (currency, URL, etc.).

Atrributes include:

- value (Object): merge field that references Salesforce field in the form of {!object.field}

Output, Table and List Components <apex:outputLink>

This tag creates a link to a URL.

Like its HTML equivalent anchor tag, the body of an outputLink is the text or image that displays as the link.

Attributes include:

- value (String): the URL used for the link
  - Use with the $Page expression to access other Visualforce pages
- title (String): the text displayed next to the output link
- Coords (Integer): the position and shape of the hot spot for the link if used with an image

Output, Table and List Components <apex:param>

This is used as a child tag that provides a name/value pair parameter for its parent component. It can be used with:

- outputLink

Attributes include:

- name (String): the name of this parameter
- value (Object): the value for this parameter

Output, Table and List Components <apex:outputPanel>

This tag defines a set of content that is grouped together, often for the purpose of doing partial-page refreshes using AJAX (more later).

Attributes include:

- style (String): the style used to display the component
- layout (String): the layout style for the panel which is either block, inline, or none. (inline)
  - For those familier with HTML:
    - block: generates an HTML div tag
    - inline: generates an HTML span tag

Output, Table and List Components <apex:outputText>

This tag simply displays text that can be formatted using a style sheet.

- Text can be used with nested param tags to dynamically insert data

Attributes include:

- value (String): the text that is displayed
- title (String): the text that is displayed next to the output link
- style (String): the style used to display the component
- escape (Boolean): specifies whether sensitive HTML or XML characters should be escaped. (true)

Output, Table and List Components <apex:outputText>

Here's a dynamic text example using the param tag:

1. <apex:outputText style="font-style:italic" value="This is {0} text with {1}.">
2. </apex:param value="my"/>
3. </apex:param value="arguments"/>
4. </apex:outputText>

- The example above renders the following HTML:

 This is my text with arguments.

Output, Table and List Components <apex:outputText>

The example creates a table by iterating over a set of data using the default Salesforce style sheet.

It can be used within either a pageBlock or a pageBlockSection.

- Use the column tag to specify the table's columns
  - If you specify a Salesforce object for a column, the associated label is automatically used in the header
  - You can alternatively use the <apex:dataTable> or <apex:dataList> tags if the data is not being supplied by a Salesforce object or you wish to use a different style sheet

Attributes include:

- columns (Integer): number of columns in the table
- value (Object): references the collection of data, usually in the form of a merge field that references Salesforce object in the form of {!object}

Output, Table and List Components <apex:dataTable>

This tag creates an HTML table by iterating over a set of data.
Use the column tag to specify tag for the table's columns.
Attributes include:

- columns (Integer): number of columns in the table
- value (Object): references the collection of data, usually in the form of a merge field that references Salesforce object in the form of {!object}
- Rows (Integer): the number or rows to display. (0 = all rows)

Output, Table and List Components <apex:column/>

This tag creates the columns for a table.

- It can be used within either a <apex:pageBlockerTable> or <apex:dataTable>

Attributes include:

- headerValue (String): the text that should be displayed in the column header. Overrides the default label if the column is a Salesforce field
- value (Object): the contents for this column, usually in the form of a merge field that references Salesforce object in the form of {!object.field}
- Width (Integer): the number of pixels for the width of this column
- rowspan and colspan (Integer): the number or rows or columns spanned by this data

Demo

Display data in Tables in an Inline Page

Output, Table and List Components
Flash

Flash code is compiled into .swf files that can be embedded into Web pages and then played using the adobe flash player.

In HTML, including a flash file might look like this:

```
1. <object data="movie.swf" type="application/x-shockwave-flash"
   width="300" height="500">
2. <param name="nameofParameter1" value="valueofParameter1"/>
3. <param name="nameofParameter1" value="valueofParameter1"/>
4. </object>
```

- Notice that this tag includes the name of the file, the dimensions of how it should be displayed, and a list of name/value pairs of parameters
  - You may not have access to the flash source code, so it is important that you understand the name/value pairs that are expected

Output, Table and List Components
<apex:flash>

In Visualforce, flash tags are used instead to call .swf files.
Attributes include:

- src (String): the name and location of the .swf file, generally within the static resources
- Height & width (Integer): the dimensions of the output
- flashVars (String): name/value pairs separated by an ampersand

Example:

```
<apex:flash
flashVars="startValue={!field}&min=1&max=5&snapInterval=1&callbackFunction=updateHiddenValue&passthroughId={!$Component.theForm.reviewBlock.scores.hiddenfieldId)&bgColor=#F3F3EC" src={!$Resource.flashSlider}"
height="40" width="175" />
```

Output, Table and List Components <apex:facet>

A facet is a child of another Visualforce component that overrides an area of the rendered parent with the contents of the facet.

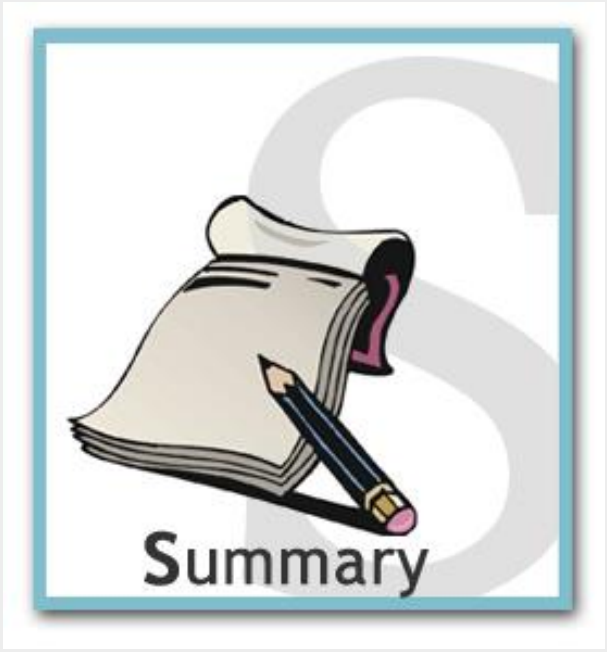
Facet tags can be used with a varity of other component tags to provide or override headers, footers, and captions to other items.

Attributes include:

- name (String): describes if the facet is a header, footer, or caption

Summary

Form Components
Output, Table and List Components



Module Review

Which tags should be used to automatically bring in the Salesforce label and default widget for a field?

What does the <apex:flash> tag do?

Where can the <apex:column/> tag be used?