

# Visualforce Tutorials – Learn Salesforce

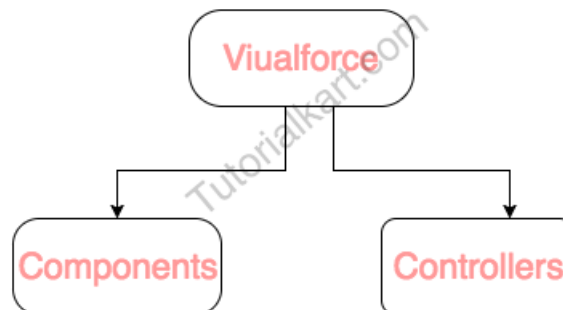
## Visualforce page Coding

### Visualforce Tutorials – Learn Salesforce Visualforce page Coding

Are you searching for Visualforce tutorials? Are you ready to learn Salesforce Visualforce page from the basics? This is the best place to Learn Visualforce. Let us learn What is Visualforce, about Visualforce components and controllers.

### What is Salesforce Visualforce?

Visualforce is the [Markup Language](#) (Tag based language) similar to HTML and XML which was developed by Salesforce.com. It contains tags and controllers where tags are also called as Components. Visualforce uses XML like syntax to create front end design pages and it uses APEX as the backend to implement business logic.



Visualforce and Apex are the mainly used to create Visualforce pages upto 15Mb that are compatible with mobile devices and browsers. VF page contains Tags, HTML and JavaScript.

### What is a Visualforce Page?

Visualforce pages have some unique components that make them easy to create feature rich user interfaces for Force.com. They use standard web technologies such as HTML, JavaScript, and CSS. Pages are composed on the server and not the client. In this Visualforce [developer tutorial](#), we have to understand about Components and Controllers.

### Visualforce Components.

Visualforce components or tags are small, reusable pieces of functionalities which are used build high quality visualforce page. In Salesforce.com, we have two types of components they are Standard Components and Custom Components.

## Standard Components or tags.

Salesforce.com by default provides 60 standard User Interface components. These are the components provided by Salesforce and these are started with `<Apex: Tagname> </Apex: Tagname>`. It creates perfect look and feel of the Salesforce without using additional CSS and HTML.

## Custom Components or Custom tags

- There are the components created by users and starts with `<c : Tagname> </c : Tagname>`.

## What is a Visualforce controller?

A Visualforce controller is a set of instructions that specify what happens when a user interacts with components or tags specified in associated Visualforce markup, such as when a user clicks a button or link it performs some logic. In Salesforce, we have three types of controllers Standard Controllers, Custom controllers and Extension controllers.

## Standard Controllers

Standard Controllers are used to customize both Standard objects and Custom objects. This controllers have access to Salesforce database and automatically adjusts to the type of the pages based on where it is called.

- We can use Standard controllers for custom objects and Standard objects.
- They provide standard functionality like save, delete and create records.
- Some standard object don't have standard controllers.

```
<apex:page standardController="Media__c">
```

## Custom Controller

Custom controller is a Apex class that implements all the login of visualforce page without leveraging the standard functionality. These are used to create wizards, rich UI with complex data sets and so on.

```
<apex: page controller="CustomApexClass">
```

- Custom Apex Class is an Apex class that controls the Visualforce Actions and Variables.

## Extension Controllers

Extension controller in visualforce is an Apex class that is used to add and extend the functionality to Standard controller and Custom controller. It is also used to overwrite the methods.

- This controller can override existing standard links and a button.

```
<apex: page standardController="Media__c"  
    extensions="CustomApexClass, SecondCustomApexClass">
```

CustomApexClass,SecondCustomApexClass are Apex classes and we can add multiple extension on a Vf page using a comma separated value.

1. [What is Visualforce?](#)
2. [Difference between Salesforce.com, Force.com and Salesforce1.](#)
3. [Why should we learn Visualforce?](#)
4. [How to create our first Visualforce page?](#)
5. [Building Visualforce page using MVC\(Model View Controller\) model.](#)
6. [VF Standard Controllers.](#)
7. [Visualforce Custom Controllers.](#)
8. [Visualforce Controller extensions.](#)
9. [Visualforce Components.](#)
  1. [apex:page.](#)
  2. [apex:pagemessage.](#)
  3. [apex:Form.](#)
  4. [apex:Commandbutton.](#)
  5. [apex:pageblockbutton.](#)
  6. [apex:pageblockSection.](#)
  7. [apex:pageblockSectionItem.](#)

### Visualforce Tutorials.

#### ⇒ [What is Visualforce?](#)

- ◆ [Difference between Salesforce.com, Force.com and Salesforce1.](#)
- ◆ [Why should we learn Visualforce?.](#)
- ◆ [Creating first visualforce page](#)

### Visualforce basics

- ◆ [Building Visualforce page using MVC Model.](#)
- ◆ [Visualforce Standard Controllers](#)

- ◆ Visualforce Custom Controllers
- ◆ Visualforce Controller extensions.
- ◆ Visualforce standard list controllers.

### **Visualforce components.**

- ◆ apex:page.
- ◆ apex:pagemessage.
- ◆ apex:Form
- ◆ apex:pageblock.
- ◆ apex:Commandbutton.
- ◆ apex:pageblockbuttons.
- ◆ apex:pageblockSection.
- ◆ apex:pageblockSectionItem.
- ◆ apex:detail.
- ◆ apex:tab.
- ◆ apex:tabpanel.
- ◆ apex:panelbar
- ◆ apex:panelbarItem.
- ◆ apex:facetname

### **Visualforce - Input Components.**

- ◆ apex:inputText.
- ◆ apex:inputSecret.
- ◆ apex:inputHidden.
- ◆ apex:inputCheckbox
- ◆ apex:inputTextArea.
- ◆ apex:selectList
- ◆ apex:selectoption
- ◆ apex:inputField
- ◆ apex:selectRadio
- ◆ apex:selectCheckBoxes

### **Enchancing your Organization.**

**Using Developer Console.**

**Unit Test**