

- **Question 1. Onchange Event Does Not Work With <apex:actionsupport> In Ie9. How To Resolve This Error?**

**Answer :**

If we add the Header on Visualforce page then it creates lots of problem in IE9. I think there are few java-script library loaded by Header of Salesforce which makes IE9 compatible. So the best solution is to enable the Header by using "showHeader=true" in Apex page.

- **Question 2. If Ie9 Is Not Working With Your Custom Visualforce Page Then How To Tell Your Visualforce Code To Run In Ie8 Compatibility Mode?**

**Answer :**

Add following metatag to pages:

```
<meta http-equiv="X-UA-Compatible" content="IE=EmulateIE8" />
```

- [Web Services Interview Questions](#)

**Question 3. It May Happen That Above Tips Will Not Work As Lots Of Time The Page Header Already Set. Then, How To Achieve Same Result Using Apex?**

**Answer :**

Add below line of code in Apex (Constructor)

```
Apexpages.currentPage().getHeaders().put('X-UA-Compatible', 'IE=8');
```

- **Question 4. How To Display The Formatted Number / Date In Visual Force? Which Component Should Be Used?**

**Answer :**

**Use component "<apex:outputText>"**.

**Example :** Format the number into currency.

```
<apex:outputtext value="{0, number, 000,000.00}">
```

```
  <apex:param value="{!valFromController}" />
```

```
</apex:outputtext>
```

OR

```
<apex:outputtext value="{0, number, ###,###.00}">
```

```
  <apex:param value="{!valFromController}" />
```

```
</apex:outputtext>
```

- [Web Services Tutorial](#)

**Question 5. You Want To Display The Encrypted Field On Visualforce And You Are Using Component Apex:outputtext. Will It Work For Encrypted Fields?**

**Answer :**

Encrypted custom fields that are embedded in the <apex:outputText> component display in clear text. The <apex:outputText> component doesn't respect the View Encrypted Data permission for users. To prevent showing sensitive information to unauthorized users, use the <apex:outputField> tag instead.

- [AJAX Interview Questions](#)

**Question 6. Will Below Query Work? Explain.**

**select Count(id), Name, Address\_\_c From Opportunity Group By Name**

**Answer :**

**Above query will throw an error.**

**Explanation:** In Group by clause the columns selected must be either used in Group by clause or in aggregate functions. The Name field is neither used in aggregate methods and in group by clause and hence will result in error "Malformed Query".

- **Question 7. Explain Difference In Count() And Count(fieldName) In Soql.?**

**Answer :**

**COUNT()**

- COUNT() must be the only element in the SELECT list.
- You can use COUNT() with a LIMIT clause.
- You can't use COUNT() with an ORDER BY clause. Use COUNT(fieldName) instead.
- You can't use COUNT() with a GROUP BY clause for API version 19.0 and later. Use COUNT(fieldName) instead.

- **COUNT(fieldName)**

- You can use COUNT(fieldName) with an ORDER BY clause.
- You can use COUNT(fieldName) with a GROUP BY clause for API version 19.0 and later.

- [AJAX Tutorial Salesforce Admin Interview Questions](#)

**Question 8. How To Write The “where” Clause In Soql When Group By Is Used?**

**Answer :**

We cannot use the “Where” clause with Group By instead we will need to use the “Having Clause”.

**Example:**

Get all the opportunity where more than one record exists with same name and name contains “ABC”.

**SELECT COUNT(Id) , Name FROM Opportunity GROUP BY Name  
Having COUNT(Id) > 1 AND Name like '%ABC%'**

- **Question 9. Let’s Consider That The First Component In Vf Page Is The Datepicker. In That Case Whenever The Page Loads, Salesforce Auto Focus The First Component Resulting In Datepicker Onfocus Event. Because Of This The Datepicker Component Opens Automatically. How We Can Avoid This?**

**Answer :**

On load event, write the javascript code to autofocus any other field or any other non-visible component.

**Example :**

```
<span id="focusDistraction"></span>
```

```
<script type="text/javascript">
```

```
    /* prevent autopup of the date inputfield by the default focus  
    behavoir */
```

```
    window.onload=function() {
```

```
        document.getElementById('focusDistraction').focus();
```

```
    }
```

```
</script>
```

- [Salesforce Developer Interview Questions](#)

**Question 10. How To Force Lead Assignment Rule Via Apex While Updating Or Adding The Lead?**

**Answer :**

To enforce Assignment Rules in Apex you will need to perform following steps:

1. Instantiate the “Database.DMLOptions” class.

2. Set the “useDefaultRule” property of “assignmentRuleHeader” to True.

3. Finally call a native method on your Lead called “setOptions”, with the Database.DMLOptions instance as the argument.

// to turn ON the Assignment Rules in Apex

```
Database.DMLOptions dmlOptn = new Database.DMLOptions();
```

```
dmlOptn.assignmentRuleHeader.useDefaultRule = true;
```

```
leadObj.setOptions(dmlOptn);
```

- [Javascript Advanced Tutorial](#)

Question 11. Access Custom Controller-defined Enum In Custom Component?

Answer :

We cannot reference the enum directly since the enum itself is not visible to the page and you can't make it a property.

Example:

Apex class:

```
global with sharing class My_Controller {  
    public Case currCase {get; set; }  
    public enum StatusValue {RED, YELLOW, GREEN}  
    public StatusValues getColorStatus() {  
        return StatusValue.RED; //demo code - just return red  
    }  
}
```

Visualforce page:

```
<apex:image url='stopsign.png' rendered="{!colorStatus ==  
StatusValue.RED}"/>
```

Above code snippet will throw error something like “Save Error: Unknown property ‘My\_Controller.statusValue’”

Resolution:

Add below method in Apex Controller:

```
public String currentStatusValue { get{ return  
getColorStatus().name(); }}
```

and change Visualforce code to

```
<apex:image url='stopsign.png' rendered="{!currentStatusValue ==  
'RED'}" />
```

- [Javascript Advanced Interview Questions](#)

**Question 12. What Is The Need Of “custom Controller” In Visualforce As Everything Can Be Done By The Combination Of Standard Controller + Extension Class.?**

**Answer :**

- Sharing setting is applied on standard object/extension by default; In case we don't want to apply sharing setting in our code then Custom controller is only option.
- It is possible that the functionality of page does not required any Standard object or may require more than one standard object, then in that case Custom controller is required.

- [Web Services Interview Questions](#)

**Question 13. In Class Declaration If We Don't Write Keyword “with Sharing” Then It Runs In System Mode Then Why Keyword “without Sharing” Is Introduced In Apex?**

**Answer :**

Let's take example, there is classA declared using “with sharing” and it calls classB method. classB is not declared with any keyword then by default “with sharing” will be applied to that class because originating call is done through classA. To avoid this we have to explicitly define classB with keyword “without sharing”.

- [XHTML Tutorial](#)

**Question 14. If User Doesn't Have Any Right On Particular Record And Have Only Read Level Access At Object Level. Can He Change The Record Owner?**

**Answer :**

Yes. In profile, there is setting for “Transfer Record”.

- **Question 15. In Which Scenario Share Object “mycustomobject\_\_share” Is Not Available/created For Custom Object “mycustomobject” ?**

**Answer :**

The object's organization-wide default access level must not be set to the most permissive access level. For custom Objects, that is Public Read/Write.

- [XHTML Interview Questions](#)

**Question 16. How To Get The Picklist Value In Apex Class?**

**Answer :**

Using Dynamic apex, we can achieve this. On object of type picklist, call `getDescribe()`. Then call the `getPicklistValues()` method. Iterate over result and create a list. Bind it to `<apex:selectOptions>`.

**Code Example:**

Let's say we have a custom object called `OfficeLocation__c`. This object contains a picklist field `Country__c`.

The first thing we need to do, within our controller is use the `getDescribe()` method to obtain information on the `Country__c` field:

```
Schema.DescribeFieldResult fieldResult =  
OfficeLocation__c.Country__c.getDescribe();
```

We know that `Country__c` is a picklist, so we want to retrieve the picklist values:

```
List<Schema.PicklistEntry> ple = fieldResult.getPicklistValues();
```

The only thing left for us to do is map the picklist values into an `<apex:selectOptions>` tag can use for display. Here is the entire method from our controller to do this:

```
public List<SelectOption> getCountries()  
{  
    List<SelectOption> options = new List<SelectOption>();  
    Schema.DescribeFieldResult fieldResult =  
OfficeLocation__c.Country__c.getDescribe();  
    List<Schema.PicklistEntry> ple = fieldResult.getPicklistValues();  
    for( Schema.PicklistEntry f : ple)  
    {  
        options.add(new SelectOption(f.getLabel(), f.getValue()));  
    }  
    return options;  
}
```

With our controller logic all complete, we can call the `getCountries()` method from our Visualforce page, and populate the `<apex:selectList>` tag:

```

<apex:selectList id="countries"
value="{!Office_Location__c.Country__c}"
size="1" required="true">
  <apex:selectOptions value="{!countries}"/>
</apex:selectList>

```

- Question 17. What Are The Types Of Controller In Visualforce?

Answer :

There are basically two types of Controller in Visual force page.

1. Standard Controller
2. Custom Controller

- [Server Load Balancing Interview Questions](#)

Question 18. How Many Controllers Can Be Used On Single Vf Page?

Answer :

Only one controller can be used salesforce. Other than them, Controller extension can be used.

There may be more than one Controller extension.

Example:

```

<apex:page standardController="Account"
extensions="ExtOne,ExtTwo" showHeader="false">
<apex:outputText value="{!foo}" />
</apex:page>

```

if ExtOne and ExtTwo, both have the method getFoo() then the method of ExtOne will be executed.

A controller extension is any Apex class that contains a constructor that takes a single argument of type ApexPages.StandardController or CustomControllerName, where CustomControllerName is the name of a custom controller that you want to extend.

- [AJAX Interview Questions](#)

Question 19. Explain System.runas()?

Answer :

Generally, all Apex code runs in system mode, and the permissions and record sharing of the current user are not taken into account.

The system method, System.runAs(), lets you write test methods that change user contexts to either an existing user or a new user.

All of that user's record sharing is then enforced. You can only use

runAs in a test method. The original system context is started again after all runAs() test methods complete.

Example :

```
System.runAs(u) {  
    // The following code runs as user 'u'  
    System.debug('Current User: ' + UserInfo.getUserName());  
    System.debug('Current Profile: ' + UserInfo.getProfileId()); }  
    // Run some code that checks record sharing  
}
```

- Question 20. Explain Test.setpage().?

Answer :

It is used to set the context to current page, normally used for testing the visual force controller.

- [Oracle Apex Interview Questions](#)

Question 21. How To Round The Double To Two Decimal Places In Apex?

Answer :

Decimal d = 100/3;

Double ans = d.setScale(2);

- Question 22. In How Many Ways We Can Invoke The Apex Class?

Answer :

1. Visualforce page
2. Trigger
3. Web Services
4. Email Services

- Question 23. Can We Create Master Detail Relationship On Existing Records?

Answer :

No. As discussed above, first we have to create the lookup relationship then populate the value on all existing record and then convert it.

- [Visualforce Interview Questions](#)

Question 24. How Validation Rules Executed? Is It Page Layout / Visualforce Dependent?

Answer :



The validation rules run at the data model level, so they are not affected by the UI. Any record that is saved in Salesforce will run through the validation rules.

- [Salesforce Admin Interview Questions](#)

**Question 25. What Is The Difference Between Database.insert And Insert?**

**Answer :**

insert is the DML statement which is same as database.insert.

However, database.insert gives more flexibility like rollback, default assignment rules etc. we can achieve the database.insert behavior in insert by using the method setOptions(Database.DMLOptions)

**Important Difference:**

- If we use the DML statement (insert), then in bulk operation if error occurs, the execution will stop and Apex code throws an error which can be handled in try catch block.
  - If DML database methods (Database.insert) used, then if error occurs the remaining records will be inserted / updated means partial DML operation will be done.
- **Question 26. What Is The Scope Of Static Variable?**

**Answer :**

When you declare a method or variable as static, it's initialized only once when a class is loaded. Static variables aren't transmitted as part of the view state for a Visualforce page.

Static variables are only static within the scope of the request. They are not static across the server, or across the entire organization.

- **Question 27. Other Than Soql And Sosl What Is Other Way To Get Custom Settings?**

**Answer :**

Other than SOQL or SOSL, Custom settings have their own set of methods to access the record.

**For example:** If there is custom setting of name ISO\_Country,  
ISO\_Country\_\_c code = ISO\_Country\_\_c.getInstance('INDIA');  
//To return a map of data sets defined for the custom object (all records in the custom object), //you would use:  
Map<String,ISO\_Country\_\_c> mapCodes = ISO\_Country\_\_c.getAll();

```
// display the ISO code for India
System.debug('ISO Code: '+mapCodes.get('INDIA').ISO_Code__c);
//Alternatively you can return the map as a list:
List<String> listCodes = ISO_Country__c.getAll().values();
```

- [Salesforce Developer Interview Questions](#)

**Question 28. What Happens If Child Have Two Master Records And One Is Deleted?**

**Answer :**

**Child record will be deleted.**

- **Question 29. What Is Difference In Render, Rerender And Renderas Attributes Of Visualforce?**

**Answer :**

**render** – It works like “display” property of CSS. Used to show or hide element.

**rerender** – After Ajax which component should be refreshed – available on commandlink, commandbutton, actionsupport etc.

**renderas** – render page as pdf, doc and excel.

- **Question 30. How To Get The List Of All Available Subject In Salesforce Database Using Apex (dynamic Apex)?**

**Answer :**

```
Map<String, Schema.SObjectType> m =
Schema.getGlobalDescribe();
```

- **Question 31. How To Create Instance Of Subject Dynamically?**  
**Normally The Subject Is Created Like “account A = New Account();”.**  
**But If You Are In Situation That You Don’t Know Which Subject Is Going To Be Instantiated? Means It Will Be Decided At Runtime, How You Will Handle It?**

**Answer :**

```
public SObject getNewSobject(String t){
    // Call global describe to get the map of string to token.
    Map<String, Schema.SObjectType> gd =
Schema.getGlobalDescribe();
    // Get the token for the subject based on the type.
    Schema.SObjectType st = gd.get(t);
    // Instantiate the subject from the token.
```

```

        Subject s = st.newSubject();
        return s;
    }

```

- **Question 32. How To Get All The Fields Of Subject Using Dynamic Apex?**

**Answer :**

```

Map<String, Schema.SObjectType> m =
Schema.getGlobalDescribe() ;
Schema.SObjectType s = m.get('API_Name_Of_SObject') ;
Schema.DescribeSObjectResult r = s.getDescribe() ;
Map<String,Schema.SObjectField> fields = r.fields.getMap();

```

- **Question 33. How To Get All The Required Fields Of Subject Dynamically?**

**Answer :**

**There is no direct property available in Apex dynamic API to represent the required field. However there is another way to know about it.**

**If any fields have below three properties then it is mandatory field.**

- If it is Creatable
  - If it is not nillable and
  - If it does not have any default value
- ```

Map<String, Schema.SObjectType> m = Schema.getGlobalDescribe()
;
Schema.SObjectType s = m.get(so.apiName) ;
Schema.DescribeSObjectResult r = s.getDescribe() ;
Map<String,Schema.SObjectField> fields = r.fields.getMap() ;
for(String f : fields.keySet())
{
    Schema.DescribeFieldResult describeResult =
fields.get(f).getDescribe();
    if( describeResult.isCreateable() && !describeResult.isNillable() &&
!describeResult.isDefaultedOnCreate() )
    {
        //This is mandatory / required field
    }
}

```

```
}  
}
```

- [Javascript Advanced Interview Questions](#)

**Question 34. What Is Property In Apex? Explain With Advantages.?**

**Answer :**

Apex mainly consists of the syntax from the well known programming language Java. As a practice of encapsulation in java we declare any variable as private and then create the setters and getters for that variable.

```
private String name;  
public void setName(String n)  
{  
    name = n;  
}  
public String getName()  
{  
    return name;  
}
```

However, the Apex introduced the new concept of property from language C# as shown below:

```
public String name {get; set;}
```

As we can see how simple the code is and instead of using nearly 8 to 11 lines all done in 1 line only. It will be very useful when lots of member is declared in Apex class. It has another advantage in “number of lines of code” limit by salesforce which will drastically reduced.

- **Question 35. What Is The Controller Extension?**

**Answer :**

Any apex class having a public constructor with Custom Controller or Standard Controller object as a single argument is known as controller extension.

- **Question 36. Explain The Need Or Importance Of The Controller Extension.?**

**Answer :**

Controller extension is very useful and important concept

introduced by the salesforce recently. It gives the power to programmer to extend the functionality of existing custom controller or standard controller.

A Visualforce can have a single Custom controller or standard controller but many controller extensions.

We can say that the custom extension is the supporter of custom or standard controller.

**Consider one example:** If there is one controller written and used by the multiple visualforce pages and one of them needs some extra logic. Then instead of writing that logic to controller class (Which is used by many visualforce pages) we can create a controller extension and apply to that page only.

- [XHTML Interview Questions](#)

**Question 37. How To Read The Parameter Value From The Url In Apex?**

**Answer :**

Consider that the parameter name is "RecordType".

String recordType =

`Apexpages.currentPage().getParameters().get('RecordType');`

- **Question 38. If One Object In Salesforce Have 2 Triggers Which Runs "before Insert". Is There Any Way To Control The Sequence Of Execution Of These Triggers?**

**Answer :**

Salesforce.com has documented that trigger sequence cannot be predefined. As a best practice create one trigger per object and use comment blocks to separate different logic blocks. By having all logic in one trigger you may also be able to optimize on your SOQL queries.

- **Question 39. What Is The Difference Between Trigger.new And Trigger.old In Apex – Sfdc?**

**Answer :**

**Trigger.new:**

Returns a list of the new versions of the sObject records

Note that this sObject list is only available in insert and update triggers

i.e., Trigger.new is available in before insert, after insert, before update and after update

In Trigger.new the records can only be modified in before triggers.

Trigger.old:

Returns a list of the old versions of the sObject records

Note that this sObject list is only available in update and delete triggers.

i.e., Trigger.old is available in after insert, after update, before delete and after update.

- **Question 40. How To Restrict Any Trigger To Fire Only Once?**

**Answer :**

Triggers can fire twice, once before workflows and once after workflows.

“The before and after triggers fire one more time only if something needs to be updated, If the fields have already been set to a value, the triggers are not fired again.”

**Workaround:**

```
public class HelperClass {  
    public static boolean firstRun = true;  
}  
trigger affectedTrigger on Account (before delete, after delete, after undelete) {  
    if(Trigger.isBefore){  
        if(Trigger.isDelete){  
            if(HelperClass.firstRun){  
                Trigger.old[0].addError('Before Account Delete Error');  
                HelperClass.firstRun=false;  
            }  
        }  
    }  
}
```

- [Server Load Balancing Interview Questions](#)

**Question 41. What Are Global Variables Explain With Examples?**

**Answer :**

Global variables are the variables used to reference the general

information about the current user or your organization on a page.

**Example:**

Global variables must be referenced using Visualforce expression syntax to be evaluated, for example, `{!$User.Name}`.

List of available global variables are as below:

1. `$Action`
2. `$Api`
3. `$Component`
4. `$ComponentLabel`
5. `$CurrentPage`
6. `$Label`
7. `$Label.Site`
8. `$ObjectType`
9. `$Organization`
10. `$Page`
11. `$Profile`
12. `$Resource`
13. `$SControl`
14. `$Setup`
15. `$Site`
16. `$User`
17. `$UserRole`
18. `$System.OriginDateTime`
19. `$User.UITheme` and `$User.UIThemeDisplayed`

● **Question 42. How To Create Many To Many Relationships Between Object?**

**Answer :**

- Creating Many to Many relationship in salesforce is little tricky. You cannot create this type of relationship directly. Follow below steps to create this type of relationship.
- Create both objects which should be interlinked.
- Create one custom object (also called as junction object), which should have auto number as unique identification and create two master relationships for both objects, no need create tab for this object.

- Now on both objects, add this field as related list.
- [Oracle Apex Interview Questions](#)

**Question 43. In Which Sequence Trigger And Automation Rules Run In Salesforce.com?**

**Answer :**

**The following is the order salesforce logic is applied to a record.**

1. Old record loaded from database (or initialized for new inserts)
2. New record values overwrite old values
3. System Validation Rules
4. All Apex “before” triggers (EE / UE only)
5. Custom Validation Rules
6. Record saved to database (but not committed)
7. Record reloaded from database
8. All Apex “after” triggers (EE / UE only)
9. Assignment rules
10. Auto-response rules
11. Workflow rules
12. Escalation rules
13. Parent Rollup Summary Formula value updated (if present)
14. Database commit
15. Post-commit logic (sending email)

**Additional notes:** There is no way to control the order of execution within each group above.

- **Question 44. What Is S-control?**

**Answer :**

**S-Controls are the predominant salesforce.com widgets which are completely based on Javascript. These are hosted by salesforce but executed at client side. S-Controls are superseded by Visualforce now.**

- **Question 45. What Is A Visualforce Page?**

**Answer :**

**Visualforce is the new markup language from salesforce, by using which, We can render the standard styles of salesforce. We can still use HTML here in Visualforce. Each visualforce tag always begins with “apex” namespace.**



All the design part can be accomplished by using Visualforce Markup Language and the business logic can be written in custom controllers associated with the Page.

- Question 46. Will Visual Force Still Supports The Merge Fields Usage Like S-control?

Answer :

Just like S-Controls, Visualforce Pages support embedded merge fields, like the `{!$User.FirstName}` used in the example.

- Question 47. What Are Merge Fields? Explain With Example?

Answer :

Merge fields are fields that we can put in Email templates, mail merge templates, custom link or formula fields to incorporate values from a record.

Example: `{!CustomObject.FieldName__c}`

- Question 48. Where To Write Visualforce Code?

Answer :

You can write the code basically in 3 ways.

1. setup->App Setup->Develop->Pages and create new Visualforce page.

2. Setup -> My Personal Information -> Personal Information -> Edit check the checkbox development mode. When you run the page like this, <https://ap1.salesforce.com/apex/MyTestPage> you will find the Page editor at the bottom of the page. You can write you page as well as the controller class associated with it, there itself.

3. Using Eclipse IDE you can create the Visualforce page and write the code.

- Question 49. What Is Difference In Isnull And Isblank?

Answer :

ISNULL:

- Determines if an expression is null (blank) and returns TRUE if it is. If it contains a value, this function returns FALSE.
- Text fields are never null, so using this function with a text field always returns false. For example, the formula field `IF(ISNULL(new__c) 1, 0)` is always zero regardless of the value

in the New field. For text fields, use the ISBLANK function instead.

- Multi-select picklist fields are never null in s-controls, buttons, and email templates, so using this function with a multi-select picklist field in those contexts always returns false.
- Empty date and date/time fields always return true when referenced in ISNULL functions.
- Choose Treat blank fields as blanks for your formula when referencing a number, percent, or currency field in an ISNULL function. Choosing Treat blank fields as zeroes gives blank fields the value of zero so none of them will be null.
- Merge fields can be handled as blanks, which can affect the results of components like s-controls because they can call this function.
- When using a validation rule to ensure that a number field contains a specific value, use the ISNULL function to include fields that do not contain any value. For example, to validate that a custom field contains a value of '1,' use the following validation rule to display an error if the field is blank or any other number: `OR(ISNULL(field__c), field__c<>1)`
- **ISBLANK:**
  - Determines if an expression has a value and returns TRUE if it does not. If it contains a value, this function returns FALSE.
  - Use ISBLANK instead of ISNULL in new formulas. ISBLANK has the same functionality as ISNULL, but also supports text fields. Salesforce.com will continue to support ISNULL, so you do not need to change any existing formulas.
  - A field is not empty if it contains a character, blank space, or zero. For example, a field that contains a space inserted with the spacebar is not empty.
  - Use the BLANKVALUE function to return a specified string if the field does not have a value; use the ISBLANK function if you only want to check if the field has a value.

- If you use this function with a numeric field, the function only returns TRUE if the field has no value and is not configured to treat blank fields as zeroes.
- **Question 50. How To Schedule A Class In Apex?**  
**Answer :**  
 To invoke Apex classes to run at specific times, first implement the Schedulable interface for the class, then specify the schedule using either the Schedule Apex page in the Salesforce user interface, or the System.schedule method.  
 After you implement a class with the Schedulable interface, use the System.Schedule method to execute it. The scheduler runs as system: all classes are executed, whether the user has permission to execute the class or not.  
 The System.Schedule method takes three arguments: a name for the job, an expression used to represent the time and date the job is scheduled to run, and the name of the class.  
 Salesforce only adds the process to the queue at the scheduled time. Actual execution may be delayed based on service availability. The System.Schedule method uses the user's time zone for the basis of all schedules. You can only have 25 classes scheduled at one time.  
**Example Code:**  

```
String CRON_EXP = '0 0 * * * ?';
clsScheduledHourly sch = new clsScheduledHourly();
system.schedule('Hourly Sync', CRON_EXP, sch);
```
- **Question 51. What Are Different Apis In Salesforce.com?**  
**Answer :**  
**REST API:**  
 REST API provides a powerful, convenient, and simple REST-based Web services interface for interacting with Salesforce. Its advantages include ease of integration and development, and it's an excellent choice of technology for use with mobile applications and Web projects. However, if you have a large number of records to process, you may wish to use Bulk API, which is based on REST principles and optimized for large sets of data.

**SOAP API:**

**SOAP API provides a powerful, convenient, and simple SOAP-based Web services interface for interacting with Salesforce. You can use SOAP API to create, retrieve, update, or delete records. You can also use SOAP API to perform searches and much more. Use SOAP API in any language that supports Web services.**

**For example, you can use SOAP API to integrate Salesforce with your organization's ERP and finance systems, deliver real-time sales and support information to company portals, and populate critical business systems with customer information.**

**Chatter API:**

**Chatter API is a REST API that provides programmatic access to Chatter feeds and social data such as users, groups, followers, and files. It's used by developers who want to integrate Chatter into a variety of applications such as mobile applications, intranet sites, and third-party Web applications. Chatter API is similar to APIs offered by other companies with feeds, such as Facebook and Twitter. Its advantages include ease of integration and development.**

**Bulk API:**

**Bulk API is based on REST principles and is optimized for loading or deleting large sets of data. You can use it to query, insert, update, upsert, or delete a large number of records asynchronously by submitting batches which are processed in the background by Salesforce.**

**SOAP API, in contrast, is optimized for real-time client applications that update small numbers of records at a time. Although SOAP API can also be used for processing large numbers of records, when the data sets contain hundreds of thousands of records, it becomes less practical. Bulk API is designed to make it simple to process data from a few thousand to millions of records.**

**The easiest way to use Bulk API is to enable it for processing records in Data Loader using CSV files. This avoids the need to write your own client application.**

**Metadata API:**

**Use Metadata API to retrieve, deploy, create, update, or delete**

**customizations for your organization. The most common use is to migrate changes from a sandbox or testing organization to your production environment. Metadata API is intended for managing customizations and for building tools that can manage the metadata model, not the data itself.**

**The easiest way to access the functionality in Metadata API is to use the Force.com IDE or Force.com Migration Tool. These tools are built on top of Metadata API and use the standard Eclipse and Ant tools respectively to simplify the task of working with Metadata API. Built on the Eclipse platform, the Force.com IDE provides a comfortable environment for programmers familiar with integrated development environments, allowing you to code, compile, test, and deploy all from within the IDE itself. The Force.com Migration Tool is ideal if you want to use a script or a command-line utility for moving metadata between a local directory and a Salesforce organization.**

#### **Streaming API:**

**Use Streaming API to receive notifications for changes to data that match a SOQL query that you define.**

**Streaming API is useful when you want notifications to be pushed from the server to the client. Consider Streaming API for applications that poll frequently. Applications that have constant polling action against the Salesforce infrastructure, consuming unnecessary API call and processing time, would benefit from this API which reduces the number of requests that return no data.**

**Streaming API is also ideal for applications that require general notification of data changes. This enables you to reduce the number of API calls and improve performance.**

#### **Apex REST API:**

**Use Apex REST API when you want to expose your Apex classes and methods so that external applications can access your code through REST architecture. Apex REST API supports both OAuth 2.0 and Session ID for authorization.**

#### **Apex SOAP API:**

**Use Apex SOAP API when you want to expose your Apex methods as SOAP Web service APIs so that external applications can access**

your code through SOAP. Apex SOAP API supports both OAuth 2.0 and Session ID for authorization.

- **Question 52. How To Display Error Message On Visualforce Page?**

**Answer :**

In the Visualforce page add the tag:

```
<apex:pageMessages />
```

In the controller class add the error message where required

```
if ( requiredFieldName == null){
```

```
    ApexPages.addMessage(new
```

```
    ApexPages.Message(ApexPages.Severity.ERROR, 'Please enter a  
value in the Required Field'));
```

```
}
```

- **Question 53. What Is Visualforce View State? And What Are Best Practices To Reduce The View State Size?**

**Answer :**

Visualforce pages that contain a form component also contain an encrypted, hidden form field that encapsulates the view state of the page. This view state is automatically created, and as its name suggests, it holds the state of the page – state that includes the components, field values and controller state.

**Best Practices to reduce the view state size:**

- Minimize number of form on a page. Use apex:actionRegion instead of using 2 or more forms.
  - Refine your SOQL to only retrieve the data needed by the page.
  - All public and private data members present in Standard, Custom and Controller extensions are saved.
  - Mark any Apex variables that are not necessary to the view state as Transient. (The transient variables are not passed to view state and therefore not stored in View State)
  - Create wizards with as few pages as possible
  - Use outputLink components instead of commandLink or commandButton components where possible as they don't need to be nested in a form.
- **Question 54. What Are Custom Settings?**

**Answer :**

Custom settings are similar to custom objects and enable application developers to create custom sets of data, as well as create and associate custom data for an organization, profile, or specific user. All custom settings data is exposed in the application cache, which enables efficient access without the cost of repeated queries to the database. This data can then be used by formula fields, validation rules, Apex, and the SOAP API.

There are two types of custom settings:

#### **List Custom Settings**

A type of custom setting that provides a reusable set of static data that can be accessed across your organization. If you use a particular set of data frequently within your application, putting that data in a list custom setting streamlines access to it. Data in list settings does not vary with profile or user, but is available organization-wide. Because the data is cached, access is low-cost and efficient: you don't have to use SOQL queries that count against your governor limits.

#### **Hierarchy Custom Settings**

A type of custom setting that uses a built-in hierarchical logic that lets you “personalize” settings for specific profiles or users. The hierarchy logic checks the organization, profile, and user settings for the current user and returns the most specific, or “lowest,” value. In the hierarchy, settings for an organization are overridden by profile settings, which, in turn, are overridden by user settings.

- **Question 55. What Is Apex?**

**Answer :**

It is the in-house technology of salesforce.com which is similar to Java programming with object oriented concepts and to write our own custom logic.

- Apex is a procedural scripting language in discrete and executed by the Force.com platform.
- It runs natively on the Salesforce servers, making it more powerful and faster than non-server code, such as JavaScript/AJAX.
- It uses syntax that looks like Java
- Apex can be written in triggers that act like database stored

procedures.

- Apex allows developers to attach business logic to the record save process.

- It has built-in support for unit test creation and execution.

Apex provides built-in support for common Force.com platform idioms, including:

- Data manipulation language (DML) calls, such as INSERT, UPDATE, and DELETE, that include built-in DmlException handling

- Inline Salesforce Object Query Language (SOQL) and Salesforce Object Search Language (SOSL) queries that return lists of sObject records

- Looping that allows for bulk processing of multiple records at a time

- Locking syntax that prevents record update conflicts

- Custom public Force.com API calls that can be built from stored Apex methods

- Warnings and errors issued when a user tries to edit or delete a custom object or field that is referenced by Apex

**Note:** Apex is included in Unlimited Edition, Developer Edition, Enterprise Edition, and Database.com

Apex vs. Java: Commonalities

- Both have classes, inheritance, polymorphism, and other common OOP features.

- Both have the same name variable, expression, and looping syntax.

- Both have the same block and conditional statement syntax.

- Both use the same object, array, and comment notation.

- Both are compiled, strongly-typed, and transactional.

Apex vs. Java: Differences

- Apex runs in a multi-tenant environment and is very controlled in its invocation and governor limits.

- To avoid confusion with case-insensitive SOQL queries, Apex is also case-insensitive.

- Apex is on-demand and is compiled and executed in cloud.

- Apex is not a general purpose programming language, but is instead a proprietary language used for specific business logic functions.



- Apex requires unit testing for development into a production environment.

- **Question 56. Explain The Apex Data Manipulation Language (dml) Operations?**

**Answer :**

Use data manipulation language (DML) operations to insert, update, delete, and restore data in a database.

You can execute DML operations using two different forms:

Apex DML statements, such as:

`insertSObject[]`

Apex DML database methods, such as:

`Database.SaveResult[] result = Database.Insert(SObject[])`

While most DML operations are available in either form, some exist only in one form or the other.

The different DML operation forms enable different types of exception processing:

Use DML statements if you want any error that occurs during bulk DML processing to be thrown as an Apex exception that immediately interrupts control flow (by using `try. . .catch` blocks). This behavior is similar to the way exceptions are handled in most database procedural languages.

Use DML database methods if you want to allow partial success of a bulk DML operation—if a record fails, the remainder of the DML operation can still succeed. Your application can then inspect the rejected records and possibly retry the operation. When using this form, you can write code that never throws DML exception errors. Instead, your code can use the appropriate results array to judge success or failure. Note that DML database methods also include a syntax that supports thrown exceptions, similar to DML statements.

|  |  |
|--|--|
|  |  |
|  |  |
|  |  |

