

Salesforce Apex Tutorials – Learn Salesforce Apex Programming language

Salesforce Apex Tutorials | Apex programming Language

Salesforce Apex is the world's first on demand strongly typed programming language developed by Salesforce.com. Apex is an object oriented language similar to C# and [Java](#) that allows to implement complex business requirements and transactions on the force.com platform.

[Salesforce Apex](#) is designed to process large amounts of data to add business logic to applications and to write Controller in the [MVC architecture](#). Declarative programming and Programmatic development are the two Application development types in Salesforce.

Salesforce Apex programming language comes under Programmatic development. After developing User Interface for an application, as a [Salesforce developer](#) it is essential to add custom logic to your application and write controller code using Salesforce Apex [Programming](#) language.

What is Salesforce Apex Programming Language?

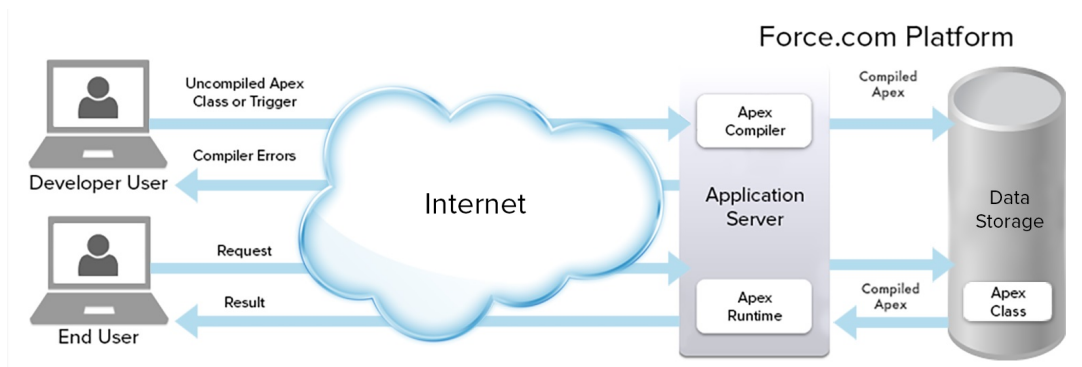
Salesforce Apex language is a multi-purpose programming language and exists in multi-tenant Environment. Every Resource that executed using Apex Programming language are effectively controlled for high quality of service to all [Salesforce developers](#).

- Apex classes and triggers can be written in developer and [sandbox](#) organization.
- Test methods is written to implement Apex code.
- The code coverage must be greater than 75%.

Salesforce Apex Programming Language Features.

1. Salesforce offers Data manipulation language (DML) with built in exception handling.
2. [Salesforce Object Query Language](#) (SOQL) and Salesforce Object Search language (SOSL) are used to query and retrieve data.
3. Apex in Salesforce has built-in record-locking mechanism. This prevents record update conflicts.
4. Salesforce Apex programming language runs on multi tenant Environment.
5. Apex code is stored in Metadata format.
6. It has syntax and variables similar to Java.
7. It provides unit testing and test execution with code coverage.
8. When we want to add web and Email services to your Application.
9. It is used to perform complex business processes.
10. When can add complex validation rules to your application.
11. Salesforce Apex can be used to add a custom logic on operations like saving a record.

How does Apex works – Working Structure of Apex.



Apex programming language runs entirely on Force.com platform. When a developer writes and Saves a code in the flat form, the code gets compiled and stores in the form of metadata in Salesforce servers. Now the end user sends request from User Interface and retrieve the results from Salesforce Servers.

How to write Apex code?

Apex programming code can be written in several development environment. We can write Apex code using [Force.com developer console](#) and using Force.com IDE.

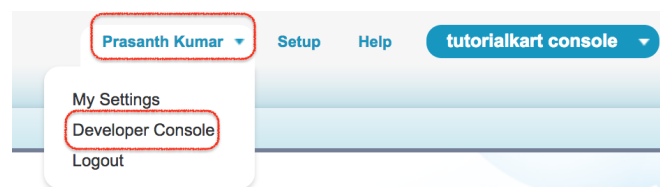
Why Force.com Developer Console is used?

Developer console in Salesforce is an integrated development environment which provides set of tools and resources that are essential to create, debug and test applications. Using Force.com [Developer Console](#), we can perform the following tasks.

1. We can add code using Source code editor.
2. Saved triggers and Classes will be compiled automatically and compiling errors will be listed.
3. Debug logs can be viewed.
4. Code coverage can be inspected and we can execute tests to specific classes.
5. Data in an organization can be retrieved using developer console.

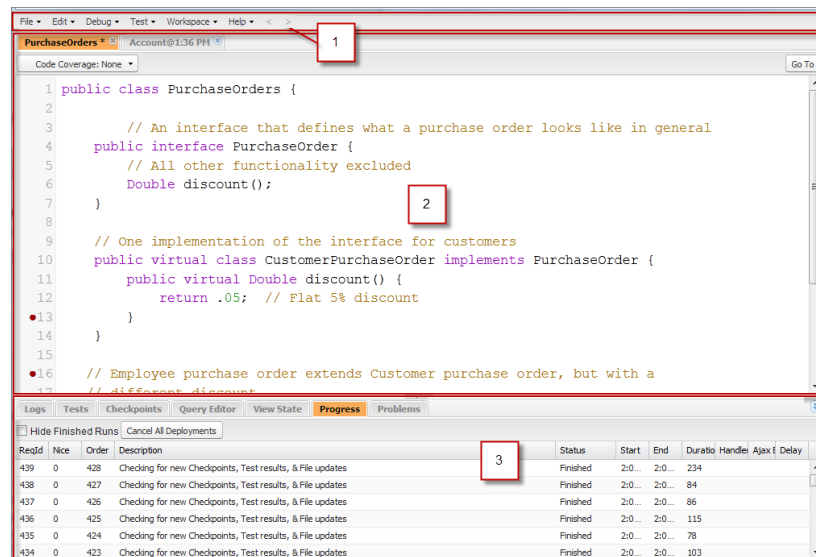
How to open Developer Console in Salesforce?

To open Developer Console in Salesforce navigate to User Menu (Name)| Developer Console.



Understanding Salesforce Developer Console Interface.

Developer Console in Salesforce included some tools which are used for coding, debugging and testing Salesforce applications.



Developer console User interface is organized into three sections.

- **Menu Bar** : In menu bar, some important menus like File, Edit, Debug, Test, Workspace and Help are present. These menus are used to manage Apex triggers, classes, can used to edit code files using Edit menu and Debug is used to manage logs and execute apex code.
- **Workspace** : Workspace section is a collections of resource with a tab for each open item. It contains some menus like Switch workspace, New Workspace, Rename Current Workspace and Workspace manager.
- **Logs, Tests and Problems Panel**: This section contains some tabs like Checkpoints tabs, Log tab, View State tab and Tests tab. These tabs are used to display saved checkpoints, to access logs in the database, to examine view state of the visualforce page request and to check Apex code coverage.

These **Salesforce Apex tutorial** is targeted for [Salesforce Developer](#) to [learn Salesforce](#) Apex programming language. These Salesforce tutorials will help to start Apex Salesforce coding for beginners. Advanced Apex Programming topics will also covered for intermediate level salesforce users. If you have used [Java Programming](#) language before then Apex Programming is easy to learn. Apex syntax is 70% similar to [Java](#).

Salesforce Apex Programming language Course – topics

What is Apex programming language, When should we use Apex Salesforce, How to write Apex code, How does Apex works, Different components in Apex code, Variable declaration, Looping Statement, [SOQL](#) (Salesforce Object Query Language), [SOSL](#) (Salesforce Object Search Language), Data Types and variables, Classes and methods in Apex Language, exception statements, Apex Triggers, apex governor limits in salesforce, Apex batch processing, Apex DML operations, Apex debugging in Salesforce, Deployment of Apex and many more.

- [SFDC Apex Interview Questions.](#)
- [Visualforce Interview Questions.](#)

Home

▸ [What is Salesforce Apex?](#)

Apex Basics

▸ [How to Enable Developing Mode in Salesforce?](#)

▸ [How to use Salesforce developer Console.](#)

▸ [Apex Data types.](#)

▸ [Apex - Variables](#)

▸ [Apex - Class](#)

▸ [Apex - Methods](#)

▸ [Apex - Constructors](#)

▸ [Apex - Scheduler](#)

▸ [What is Salesfore Batch Apex](#)

▸ [Batch Apex - Governor Limits](#)

▸ [Triggers in Salesforce](#)

▸ [Email Messages - Inbound, Outbound](#)

▸ [Apex - Interface](#)

▸ [Apex - DML statements](#)