



Salesforce Tutorial for Beginners

Introduction

Is the world of CRM overwhelming, or are you having a difficult time understanding how business data management works? Many beginners find the landscape confusing and the terminology daunting. Salesforce is the world's #1 CRM, and learning it provides a powerful and marketable skill set. This tutorial cuts through the complexity and focuses on the practical, hands-on skills you need. Ready to build your career? View our complete [Salesforce course syllabus](#) now!

Why Students or Fresher Learn Salesforce?

Reasons are as follows:

- **High Demand & Career Growth:** Companies worldwide use Salesforce, creating a huge demand for certified professionals, even at entry-level. This translates to excellent job security and a clear career path.

- **Skill Gap Advantage:** University courses often don't cover Salesforce. Learning it provides a unique, in-demand skill set that immediately makes you a preferred candidate over peers.
- **Lucrative Salaries:** Salesforce roles (Admin, Developer, Consultant) are consistently ranked among the highest-paying tech jobs for entry-level professionals.
- **Industry Agnostic:** The skills are transferable across almost all industries (Finance, Health, Retail, etc.), offering maximum career flexibility.
- **Easy Entry Point:** The platform has a low barrier to entry for beginners, especially with free learning resources like Trailhead.

Ready to land your first job? Download our free guide to top [Salesforce interview questions and answers](#)!

**Take your Knowledge
Test Report**

Check your Score



Step-by-Step Salesforce Tutorial for Beginners

A structured, hands-on approach is the best way to master Salesforce. Since Salesforce is a cloud-based platform (SaaS), there is no complex installation required. The first step is simply to set up your free learning environment.

This comprehensive Salesforce Tutorial will walk you through setting up your free Developer Edition (Dev Org) and provide step-by-step guide on the core concepts of the platform.

Part 1: Setup – Your Free Salesforce Learning Environment

The Salesforce Developer Edition (Dev Org) is your free, permanent environment to practice, build, and test applications without affecting a real company's data.

1.1. Create Your Developer Account

1. **Visit the Developer Signup Page:** Find the Sign-up page by navigating to the official Salesforce Developer website.
2. **Fill in the Details:** Give your personal information, like your Name, Email, and Company Name (you can put "Self-Employed" or "Personal Learning").
3. **Choose Your Username:** Your username must be in the format of an email address (e.g., myname@learning.com), but it doesn't have to be a real, verifiable email address. However, the Email field must be a valid email.
4. **Sign Up:** Click the "Sign me up" button.
5. **Verify Your Account:** Check the inbox of the actual email address you provided. Salesforce will send a verification email. Click the "Verify Account" link in the email.
6. **Set Your Password:** You will be prompted to set your password and a security question. Complete this step, and you'll be logged into your new Salesforce Developer Org!

1.2. First Login and Overview

Upon logging in, you'll land on the Lightning Experience interface, which is the modern standard for Salesforce.

- **App Launcher (The Waffle Icon):** Located at the top-left, this is where you can switch between different Apps (like Sales, Service, or Marketing). An App is a collection of related tabs, items, and functionality.
- **Navigation Bar (Tabs):** This bar, just below the App Launcher, shows the tabs for the currently selected App (e.g., Accounts, Contacts, Opportunities).

- **Setup Gear Icon:** Located at the top-right, this is the gateway to Customization and Administration. Clicking it opens the Setup menu, where Admins and Developers do all the building.

Part 2: Salesforce Administration (Point-and-Click)

The core of a Salesforce Administrator's job is customization. This section covers the fundamental components of the Salesforce data model and how to customize them using point-and-click tools.

2.1. Understanding the Data Model: Objects, Fields, and Records

Salesforce is built around a database, and the key components are:

- **Object:** Think of an Object as a database table. Standard Objects are pre-built by Salesforce (e.g., Account, Contact, Lead, Opportunity). Custom Objects are ones you create to store unique business data (e.g., "Student," "Product Inventory").
- **Field:** Think of a Field as a column in the database table (e.g., Name, Phone, Email, Annual Revenue).
- **Record:** Think of a Record as a row in the database table, which is a single instance of an object (e.g., "Acme Corp" is one Account Record).

2.2. Tailoring Objects (The Building Blocks)

Let's create a Custom Object called "Candidate" to track job applicants.

Step-by-Step:

1. Click the Setup *Gear Icon* → *Select Setup*.
2. In the Quick Find box, type "*Object Manager*" and select it.
3. Click the "*Create*" dropdown and select "*Custom Object*".
4. **Fill in the details:**
 - **Label:** *Candidate*

- **Plural Label:** *Candidates*
 - **Object Name:** *Candidate (auto-populated)*
 - **Record Name:** *Candidate Name* (This is the required unique identifier, typically a Text field).
5. **Check the box:** *“Launch New Custom Tab Wizard after saving this Custom Object.”*
 6. Click *“Save.”*

Creating the Custom Tab:

- The wizard automatically appears. Choose a *Tab Style* (e.g., a briefcase icon).
- Set visibility for different Profiles: for now just leave the defaults.
- Which Apps to include the new Candidates tab within, for example, Sales App.
- Click *“Save.”* You now have a new tab and object!

2.3. Adding Custom Fields (the Columns of Data)

Let's add some fields to the new object called Candidate.

Step-by-Step:

1. In the *Object Manager*, click the *Candidate* object you just created.
2. Click *“Fields & Relationships”* on the left.
3. Click *“New”*.
4. **Field 1:** Applicant Status
 - **Data Type:** Select Picklist (to give the user a predefined list of choices).
 - **Label:** Candidate Status
 - **Enter values:** Select *“Enter values, with each value separated by a new line”*.
 - **Enter values like:** New, Screening, Interviewing, Offered, Hired, Rejected
 - Click *“Next”* → *Set visibility (keep defaults)* → Click *“Next”* → *Add to relevant Page Layouts (keep defaults)* → Click *“Save.”*

5. Field 2: Expected Salary

- **Data Type:** Choose Currency.
- **Label:** Expected Salary
- Set *Length and Decimal Places* (e.g., 16 Length, 2 Decimal Places).
- Click “Next” → Set *visibility (keep defaults)* → Click “Next” → Add to *relevant Page Layouts (keep defaults)* → Click “Save & New” (to create another).

2.4. Validation Rules (Enforcing Data Quality)

Validation Rules are point-and-click logic that prevents users from saving a record if the data doesn't meet certain criteria, ensuring data quality.

Scenario: We want to ensure that if the Candidate Status is ‘Hired’, the Expected Salary field cannot be empty.

Step-by-Step:

1. In *Object Manager* navigate to the *Candidate* object.
2. Click “**Validation Rules**” on the left.
3. Click “**New**”.
4. **Rule Name:** *Prevent_Hire_Without_Salary*
5. **Active:** *Verified*.
6. **Error Condition Formula:** This is a simple formula language that evaluates to TRUE if the error should fire. We use the ISPICKVAL function for picklists and the ISBLANK function.

AND(

ISPICKVAL(Candidate_Status__c, “Hired”),

ISBLANK(Expected_Salary__c)

)

7. **Error Message:** You cannot set the Candidate Status to 'Hired' without providing an Expected Salary.
8. **Error Location:** Desired Salary
9. Click **“Save.”**

Now, test this! Go to the Candidates tab, create a new record, set the status to 'Hired', and try to save without filling the salary. You will see your error message.

Part 3: Salesforce Automation (Flow Builder)

Salesforce Flow is the powerful, visual tool used to automate complex business processes. For beginners, it's the most important skill to learn outside of basic admin setup. We will create a simple Record-Triggered Flow.

Scenario: When a Candidate record's Candidate Status is updated to 'Hired', we want to automatically create a related Contact record.

3.1. Creating the Record-Triggered Flow

- Click the Setup *Gear Icon* → *Select Setup*.
- In the Quick Find box, type “Flows”, then select Flows under Process Automation.
- Click “New Flow”.
- Choose “Record-Triggered Flow,” then click “Create.”

3.2. Setup the Flow Trigger

- **Object:** *Select Candidate* (your custom object).
- **Trigger the Flow When:** *A record is created or updated.*
- **Condition Requirements:** *All Conditions Are Met (AND)*
- **Field:** *Candidate_Status__c*

- **Operator:** Equals
- **Value:** Hired
- **When to Run the Flow for Updated Records:** *Only when a record is updated to meet the condition requirements. (This prevents the flow from running repeatedly).*
- **Optimize the Flow For:** *Actions and Related Records (Best practice for creating or updating related records).*
- Click **“Done.”**

3.3. Creating the New Contact – Action Element

1. Click the (+) icon on the path and select **“Action”** (**“Create Records”** is also acceptable).
2. **Label:** *Create Hired Contact*
3. **API Name:** *Create_Hired_Contact (auto-populated)*
4. **How to Set Record Fields:** *Using separate resources, and literal values.*
5. **Object:** *Choose Contact.*
6. **Set Field Values for the Contact:**
 - **Field 1:** *LastName* | **Value:** *{!Record.Name}* (This is a variable that takes the Candidate’s Name)
 - **Field 2 (Important):** *AccountId* | **Value:** You might skip this for simplicity now, but in a real org, you’d relate the Contact to an Account. If you don’t have an Account ID, the Contact will be created without an Account relationship.
7. Click **“Done.”**

3.4. Save and Turn On the Flow

1. Click **“Save”** in the top-right corner.
2. **Flow Label:** *Create Contact When Candidate Hired*
3. Click **“Save”** again.

4. Click the **“Activate”** button.

Test: Go to your Candidates tab, create a new Candidate record, and set the status to ‘Hired’. Save the record. Now, go to the Contacts tab and you should see a new Contact created with the same name!

Part 4: Reporting and Dashboards

Reporting is how you turn raw data into actionable business intelligence. It’s the final output of all your customization efforts.

4.1. Creating a Simple Report

Scenario: Create a report to view all Candidates with the status of ‘Interviewing’.

1. Click the **App Launcher** → **Select “Reports”**.
2. Click **“New Report”**.
3. **Choose Report Type:** In the Quick Find, search for and select your custom report type: **“Candidates”**.
4. Click **“Start Report.”**
5. **Filter:** Filter what data is seen.
 - Click on the **“Filters”** tab.
 - By default, it shows **“Show Me: All Candidates”** and **“Date Field: Create Date, All Time.”** Leave those for now.
 - Click **“Add Filter.”**
 - **Field:** *Candidate Status*
 - **Operator:** *equals*
 - **Value:** *Interviewing*
6. **Outline (Columns):** Go to the **“Outline”** tab and drag the fields you want to see into the Columns section, like *Candidate Name, Expected Salary, and Owner*.
7. Click **“Run.”** The report updates to show only the Candidates currently in the ‘Interviewing’ stage.

8. Click **“Save.”** **Report Name:** *Candidates in Interviewing Stage*.

4.2. Creating a Dashboard Component

A Dashboard is the visualization of several reports in one place.

1. Click the **App Launcher** → Select **“Dashboards”**.
2. Click **“New Dashboard.”**
3. Name it, for example, Hiring Dashboard. Click **“Create.”**
4. Click **“+ Component”**.
5. **Search for and select your saved report:** *Candidates in Interviewing Stage*.
6. **Display As:** Select a chart type, for example **Horizontal Bar Chart**.
7. **Y-Axis:** *Candidate Status*
8. **X-Axis:** *Record Count*. The number of candidates.
9. Click **“Add.”** You now have a visual component showing the count of your interviewing candidates!

Part 5: Introduction to Apex (Developer Path)

While much of Salesforce can be done with point-and-click tools, complex business logic requires Apex, Salesforce’s proprietary, Java-like programming language.

- **Note for Beginners:** You don’t need to learn Apex immediately, but it’s important to know what it is and where it’s used (Triggers, Custom Controllers, etc.).

Apex Trigger Sample

An Apex Trigger is procedural code that executes before or after a specific Data Manipulation Language (DML) event (e.g., insert, update, delete).

Scenario: A basic trigger that ensures the Candidate Name is always converted to uppercase upon insertion.

1. **Open Developer Console:** Click the Setup **Gear Icon** → **Select “Developer Console.”**
2. **Create an Apex Trigger:** In the Developer Console, go to **File** → **New** → **Apex Trigger**.
 - **Name:** *CandidateNameUpper*
 - **sObject:** *Candidate__c*
3. **Enter the Code:** Replace the default code in with the following:

```
trigger CandidateNameUpper on Candidate__c (before insert) {  
  
    // Check if the trigger is running on an insert event  
  
    if (Trigger.isBefore && Trigger.isInsert) {  
  
        // Iterate through all new Candidate records being inserted  
  
        for (Candidate__c newCandidate : Trigger.new) {  
  
            // Check if the name field is not empty  
  
            if (newCandidate.Name != null) {  
  
                // Set the Name field to its uppercase version  
  
                newCandidate.Name = newCandidate.Name.toUpperCase();  
  
            }  
  
        }  
  
    }  
  
}
```

4. **Save:** *Ctrl+S (Windows) or Cmd+S (Mac).*
5. **Test the Trigger:** Go back to your Salesforce interface and create a new Candidate record. Enter the name in lowercase (e.g., john doe). When you save, the record will be created with the name as JOHN DOE.

The next step in your journey is to move from simple tasks to complex, real-world scenarios.

Ready to test your skills on real-world problems? Download our free guide to advanced [Salesforce Administration and Developer challenges and solutions!](#)

Real Time Examples for Salesforce Tutorial for Learners

Learning Salesforce is best done by solving real-world business problems. Here are some common scenarios ideal for a beginner's Dev Org:

Lead-to-Opportunity Automation for Sales Teams

Problem: Sales reps waste time manually converting high-quality leads and forgetting to create follow-up tasks.

Your Solution: Implement a Record-Triggered Flow on the Lead object.

- **Condition:** Status of Lead is changed to 'Qualified'.
- **Action 1:** Automatically create and convert the Lead into a new Account, Contact, and Opportunity record.
- **Action 2:** Create a follow-up Task assigned to the Opportunity Owner, with the due date set to three days from the conversion date.

Skills Reinforced: Flow automation, Lead conversion process, Task management.

Customer Service Case Management

Problem: New customer support cases are sometimes missed if they aren't quickly assigned to a specific queue or agent based on priority.

Your Solution: Create an Assignment Rule on the Case object.

- **Rule Entry 1:** If the Case Subject contains the word "Urgent" or "System Down", assign the Case to the 'High Priority Queue'.
- **Rule Entry 2:** If the Case Origin is 'Email' and the Priority is 'Low', assign the Case to the 'Standard Support Queue'.

Skills Reinforced: Assignment Rules, Queues, Case object, Data triage.

Marketing Data Quality Validation

Problem: Marketers struggle with campaigns because contact data often has missing phone numbers or invalid email formats, leading to bounced messages.

Your Solution: Create Validation Rules on the Contact object.

- **Rule 1 (Data Requirement):** Prevent saving a Contact if the Mailing State is missing when the Mailing Country is 'USA' (using `AND(ISPICKVAL(MailingCountry, 'USA'), ISBLANK(MailingState)))`).
- **Rule 2 (Format Check):** Use a complex formula (`NOT(REGEX(Email, ".*@.*\.*"))`) to ensure the Email field follows a basic a@b.c format.

Skills Reinforced: Validation Rules, Formula Language, Data integrity, Data governance.

Apply these skills and enhance your portfolio with our [Salesforce Project Ideas for beginners](#).

FAQs About Salesforce Tutorial for Beginners

1. How do I start learning Salesforce?

Start with Trailhead, Salesforce's free, gamified online learning platform. Get a free Developer Org to practice hands-on administrative tasks like creating objects, fields, and simple automation using Flow. Focus on the Salesforce Administrator Certification path first.

2. What are the 5 core values of Salesforce?

The five core values of Salesforce are: Trust, Customer Success, Innovation, Equality, and Sustainability. These values guide the company's culture, business decisions, and commitment to stakeholders, employees, and the environment.

3. Is Salesforce easy to learn for beginners?

Yes, Salesforce is generally easy for beginners to grasp initially because its core administrative functions rely on a user-friendly, "clicks-not-code" interface (declarative tools). Mastering the full platform and complex scenarios requires dedicated, continuous effort.

4. Can I self learn Salesforce?

Self-learning is the primary path for many Salesforce professionals, due to the vast, free resources available. However, we offered structured learning modules, hands-on projects, and a global community to support your journey through our affordable [online Salesforce professional course](#).

5. How to learn Salesforce in 30 days?

You can gain a strong foundational understanding in 30 days by dedicating daily hours to a structured plan. Focus intensely on core Admin topics like Data Model, Security, and Automation (Flow). This timeframe is for building a base, not full mastery.

6. Is coding required for Salesforce?

Coding is not required for a Salesforce Administrator or functional Consultant role, which uses "clicks" (declarative tools). However, Apex code and development skills are required for the Salesforce Developer and Technical Architect roles.

7. Will AI replace Salesforce developer?

No, AI will augment, not replace, Salesforce developers. Tools like Einstein for Developers handle routine code generation and testing. Developers will shift their focus

to complex architecture, system integration, nuanced business logic, and validating AI output.

8. Can I get job in Salesforce as a fresher?

Yes, you can! Many companies hire Salesforce freshers for Admin or Jr. Consultant roles, especially if you have an Administrator Certification and demonstrable hands-on experience from personal projects on a Developer Org.

9. Can a non-technical person learn Salesforce?

Yes, Salesforce is an excellent career path for non-technical people. Roles like Salesforce Administrator or Business Analyst prioritize logical problem-solving, understanding business processes, and using the declarative (point-and-click) tools over coding.

10. Is a Salesforce job stressful?

Like most tech roles, Salesforce jobs can be stressful due to tight project deadlines, complex integrations, and the need for continuous learning (three annual releases). However, the stress level varies significantly based on the specific company and role.

Conclusion

You've taken the essential first step by understanding the core of Salesforce—from setting up your free Dev Org and building custom objects to automating processes with Flow. Remember, mastery comes through practice. Salesforce is the #1 CRM platform, and developing these skills unlocks immense career potential in sales, service, and marketing technology. Embrace the continuous learning curve! Ready to formalize your expertise and accelerate your career? Enroll in our full [**Salesforce Certification course in Chennai**](#) today!