multiverse

Contents

Apex

Apex is a programming language designed specifically for the Salesforce platform. It was created by Salesforce in 2007 and has since become an integral part of the Salesforce ecosystem. Apex allows developers to build custom functionality within Salesforce, such as custom workflows, triggers, and user interfaces.

Apex is an object-oriented language that is similar in syntax to Java. It runs on the Salesforce platform and is optimized for building applications that interact with Salesforce data. Apex includes a number of features that are specific to Salesforce, such as the ability to access and manipulate data in Salesforce's database, and the ability to create custom user interfaces using the Visualforce framework.

Since its inception, Apex has undergone several updates and improvements, including the introduction of new features such as the Lightning Component Framework and the Salesforce DX development environment. Today, Apex remains a critical tool for Salesforce developers, and is widely used to build custom applications and integrations on the platform.

The 5-day learning plan that we have outlined is designed to provide a solid foundation for learning Apex programming, including the fundamentals of Apex syntax, object-oriented programming concepts, triggers, Visualforce, and integration with external systems. By following this learning plan and continuing to practice and build projects, you can become proficient in Apex programming and unlock new opportunities as a Salesforce developer.

Learning Plan Tasks

1. Introduction to Apex
2. Object Oriented Programming in Apex
3. Apex Triggers
4. Visualforce Pages
5. Advanced Apex Topics
6. Build Something Using Apex!

#checkoutTheDocs

- **Salesforce**: Salesforce Trailhead is a free online learning platform provided by Salesforce that offers interactive tutorials on a variety of topics.
- **Salesforce**: Developer Documentation
- **Apex Academy**: Apex Academy

1. Introduction to Apex

This section covers the following topics:

- Start with the official Salesforce Trailhead tutorial for Apex programming. This will give you a solid foundation for understanding the basics of the language and how it is used in the Salesforce ecosystem.
- Complete the "Apex Basics & Database" module on Trailhead. This module will cover the fundamentals of Apex syntax, variables, and control structures, as well as how to interact with Salesforce's database.

By completing these tasks you should have a solid foundation in the basics of Apex programming, including its syntax and how to interact with Salesforce's database. You should also have completed some hands-on challenges to help reinforce your learning.

Tasks

1. Start with the official Salesforce Trailhead tutorial for Apex programming:

   ☐ Create a Trailhead account so that your progress can be saved!
   ☐ Read through the "Get Started with Apex" section, which covers an overview of Apex, how it is used in the Salesforce ecosystem, and the basics of Apex syntax and how to write your first Apex class.
   ☐ Complete the "Create a Simple Apex Class" challenge by following the instructions to create a basic Apex class and test it in a Salesforce Developer Edition org.

2. Complete the "Apex Basics & Database" module on Trailhead:

   ☐ Read through the "Introduction to Apex" section, which provides an overview of Apex and its syntax.
   ☐ Complete the "Apex Basics" unit, which covers Apex syntax, variables, and control structures. Make sure to complete the challenges as you go along.
   ☐ Complete the "DML Operations" unit, which covers how to interact with Salesforce's database using Apex. Again, make sure to complete the challenges as you go along.

2. Object-Oriented Programming in Apex

This section covers the following topics:

- Start with the "Object-Oriented Programming with Apex" module on Trailhead. This will give you a more in-depth look at the object-oriented features of the language, including classes, inheritance, and interfaces.
- Practice writing and testing your own classes in a sandbox environment.

By completing these tasks you should have a good understanding of object-oriented programming concepts and how they apply to Apex, as well as some hands-on experience writing and testing your own Apex classes.

Tasks

1. Start with the "Object-Oriented Programming with Apex" module on Trailhead:

   ☐ Read through the "Introduction to Object-Oriented Programming" section, which provides an overview of object-oriented programming concepts and how they apply to Apex.
   ☐ Complete the "Classes and Objects" unit, which covers how to define and use classes in Apex, including constructors, properties, and methods.
   ☐ Complete the "Inheritance and Polymorphism" unit, which covers how to use inheritance to create subclasses of existing classes, and how to

use polymorphism to treat objects of different classes as if they were the same type.

☐ Complete the "Interfaces" unit, which covers how to define and use interfaces in Apex, which provide a way to define a set of methods that a class must implement.

2. Practice writing and testing your own classes in a sandbox environment:

☐ Create a new Salesforce Developer Edition organization if you haven't already.

☐ Use the Apex Developer Console or a code editor such as Visual Studio Code with the Salesforce Extension Pack to write Apex classes.

☐ Write a simple Apex class that defines a custom object and methods for creating, reading, updating, and deleting records of that object.

☐ Use the Developer Console or the Apex Test Execution page to write and run tests for your class to ensure that it works as expected.

3. Apex Triggers

This section covers the following topics:

- Begin with the "Triggers" module on Trailhead. This will cover how to write Apex triggers, which are used to respond to events such as record creation, deletion, or modification in Salesforce.
- Practice writing and testing your own triggers in a sandbox environment.

By completing these tasks you should have a good understanding of how to write Apex triggers and how they are used in Salesforce, as well as some hands-on experience writing and testing your own triggers in a sandbox environment.

Tasks

1. Begin with the "Triggers" module on Trailhead:

☐ Read through the "Introduction to Triggers" section, which provides an overview of what triggers are and how they are used in Salesforce.

☐ Complete the "Trigger Events and the Order of Execution" unit, which covers the different types of trigger events and how they are processed by Salesforce, as well as the order in which triggers are executed.

☐ Complete the "Anatomy of a Trigger" unit, which covers the basic structure of a trigger in Apex and how to define trigger handlers.

☐ Complete the "Bulk Triggers" unit, which covers best practices for writing triggers that can handle large amounts of data.

2. Practice writing and testing your own triggers in a sandbox environment:

☐ Using the same Developer Edition organization that you created, write a simple trigger that performs an action in response to a specific event, such as creating a new record or updating an existing record.

☐ Use the Developer Console or the Apex Test Execution page to write and run tests for your trigger to ensure that it works correctly.

☐ Experiment with different trigger events and conditions to see how they affect the behavior of your trigger.

4. Visualforce Pages

This section covers the following topics:

- Start with the "Visualforce Basics" module on Trailhead. This will cover how to create custom user interfaces using the Visualforce framework.
- Practice building and testing your own Visualforce pages in a sandbox environment.

By completing these tasks you should have a good understanding of how to create custom user interfaces using the Visualforce framework, as well as some hands-on experience building and testing your own Visualforce pages in a sandbox environment.

Tasks

1. Start with the "Visualforce Basics" module on Trailhead:

   ☐ Read through the "Introduction to Visualforce" section, which provides an overview of what Visualforce is and how it is used in Salesforce.
   ☐ Complete the "Visualforce Pages" unit, which covers how to create Visualforce pages using markup and components, and how to control their behavior using controllers and extensions.
   ☐ Complete the "Visualforce Controllers" unit, which covers how to write Apex code to control the behavior of Visualforce pages, including retrieving and updating data from Salesforce.
   ☐ Complete the "Visualforce Components" unit, which covers how to create reusable components for use in multiple Visualforce pages.

2. Practice building and testing your own Visualforce pages in a sandbox environment:

   ☐ Using the same Developer Edition organization that you created, create a simple Visualforce page that displays data from a custom object in Salesforce.
   ☐ Write a Visualforce controller to retrieve data from Salesforce and pass it to the page.
   ☐ Use the Developer Console or the Visualforce Previewer to preview and test your page.
   ☐ Experiment with different Visualforce components and styling options to customize the look and feel of your page.

5. Advanced Apex Topics

This section covers the following topics:

- Start with the "Apex Integration Services" module on Trailhead. This will cover how to integrate your - Apex code with external systems, such as web services or other databases.
- Practice writing and testing your own Apex code to integrate with an external system.

By completing these tasks you should have a good understanding of how to integrate your Apex code with external systems using web services, callouts, and outbound

messaging, as well as some hands-on experience writing and testing your own code to integrate with an external system.

Tasks

1. Start with the "Apex Integration Services" module on Trailhead:

   ☐ Read through the "Introduction to Apex Integration" section, which provides an overview of the different ways that you can integrate your Apex code with external systems.
   ☐ Complete the "Apex Web Services" unit, which covers how to create and consume SOAP and REST web services in Apex.
   ☐ Complete the "Apex Callouts" unit, which covers how to make HTTP requests from Apex code to external systems.
   ☐ Complete the "Outbound Messaging" unit, which covers how to use outbound messages to send data from Salesforce to an external system.

2. Practice writing and testing your own Apex code to integrate with an external system:

   ☐ Identify an external system that you want to integrate with Salesforce, such as a web service or a database.
   ☐ Write Apex code to interact with the external system using one of the integration methods covered in the Trailhead module.
   ☐ Use the Developer Console or the Apex Test Execution page to write and run tests for your code to ensure that it works correctly.
   ☐ Experiment with different integration scenarios and error handling to see how they affect the behavior of your code.

Build Something with Apex!

Here are some project ideas you can work on to check your understanding of Apex programming:

- Build a custom Apex trigger that automates a specific process in Salesforce, such as updating a field on a related record when a specific field on the current record is updated.
- Create a custom Visualforce page that displays data from a related object in a visually appealing and user-friendly way. You can use custom Apex code to query and manipulate the data as needed.
- Build a custom Apex class that integrates Salesforce with an external system, such as a web service or a database. You can use this class to transfer data between Salesforce and the external system, or to perform other custom actions based on data in Salesforce.
- Create a custom Lightning component that displays data from a Salesforce object in a visually appealing and interactive way. You can use custom Apex code to query and manipulate the data as needed.
- Build a custom Apex test class that thoroughly tests the functionality of an existing Apex trigger or class. You can use this class to ensure that your code is working correctly and to catch any potential issues before they arise in production.

Next Steps

Here are some suggestions for next steps to take after you have completed your learning plan:

- Prepare for and take the Salesforce Certified Platform Developer I exam. This certification exam validates your knowledge and skills in Salesforce development, including Apex programming, Visualforce, and integration with external systems. Studying for and passing the exam can help you demonstrate your expertise to potential employers or clients.
- Join the Salesforce developer community.
- Explore other areas of Salesforce development. Apex programming is just one aspect of Salesforce development, and there are many other areas to explore, such as Lightning Web Components, Heroku, and Salesforce DX.