

Table of Contents



1. [Day1](#)
2. [Primitive data type in Salesforce](#)
3. [Non-primitive data type in Salesforce](#)
4. [Assignment](#)
5. [Classes in Salesforce](#)
6. [Access Modifiers in Salesforce](#)
7. [Variables\(States\) in Salesforce](#)
8. [Hands-On Code](#)
9. [Assignment](#)
10. [Full Video](#)
11. [Resources](#)

Day1

If you have missed the previous session here is the link –

<https://www.pantherschools.com/introduction-to-oops-using-salesforce>

Primitive data type in Salesforce

They are simple, basic data types that are built into the Salesforce platform and **cannot be broken down into smaller components**.

- **Integer:** A whole number without a decimal point.
- **Double:** A number with a decimal point or a fractional component.
- **Boolean:** A data type that can have one of two possible values – true or false.
- **Date:** A data type that represents a date.
- **Time:** A data type that represents a specific time.
- **DateTime:** A data type that represents a specific date and time.
- **String:** A sequence of characters that can include letters, numbers, and symbols.
- **ID:** A unique identifier for an object in Salesforce. (18, 15) 001GHJHGHE87734
- **Decimal:** A number with a decimal point, with up to 18 digits

Non-primitive data type in Salesforce



The non-primitive data types in Salesforce include:

- Object
 - Variables
 - Primitive
 - Name
 - Age
 - Non Primitive
 - Person Object
 - Animal Object
- Map
- List
- Set
- Apex Class
 - Variables
 - Methods

Assignment

Create variables for all the Primitive Data Types and print those with dummy values.

Classes in Salesforce

A class is nothing but it is a collection of objects and carries all the properties of an object.

We can also say that the class is the blueprint for creating the Objects.

A class can contains the followings –

- Classes
 - Access Modifiers
 - Variable
 - Methods
 - Constructors

Access Modifiers in Salesforce



Access modifiers in Salesforce are used to define the **visibility and accessibility of classes, methods, variables,** and other components **within an organization.**

There are three types of access modifiers in Salesforce:

1. **Public:** Components with the public access modifier are visible and accessible to all classes and triggers in an organization. This means that any class or trigger in the organization can access and use these components.
2. **Private:** Components with the private access modifier are only visible and accessible within the same class in which they are defined. This means that no other class or trigger in the organization can access or use these components.
3. **Protected:** Components with the protected access modifier are visible and accessible within the same class and any subclasses that extend the original class. This means that any class that inherits from the original class can access and use these components.

1. Class A (Parent)

1. Protected Variables

2. Class B (Child) extend Class A (Variable & Methods)

1. Access Protected Variables

4. **Global**

1. **Managed Package (Application)** ~= Applications in Google/Apple Store

1. Class

1. Variable

2. Method

Variables(States) in Salesforce

Variables in Salesforce are used to store and manipulate data within Apex code.

```

1 | <DataType> <VariableName> = <Value>; // Variables Declrations & Value Assi
2 | Integer myAge = 5;
3 | Decimal mySalary = 76.78K;
4 | String myName = 'John Doe';

```



Hands-On Code

```

//<access modifier> class <class name>{}
//<access modifier> <access specifier>---
public class Animal {-----

```

```

5 // <access modifier> <DataType> <VariableName> = <Value>;
6 // <access modifier> <DataType> <VariableName>;-----
7 // by default access modifier will be private for any variable
8 public String name = 'Max'; // String will always be withing Single Qu
9 Integer age = 5; -----
10
11 // Instance Variable OR Object Variable
12 //Multiple-time memory allocation depends upon the of objects
13 -----
14 // Class Variable
15 public static String address = '123 Main Street'; // One time memory a
16 -----
17 }
18
19 // Anynomus Code
20 -----
21 //<Data Type> <variable name> = <variable value>;
22 // Heap Memory -----
23 String name = 'John Doe'; // Variable Declaration & Value Assignment
24 //<Data Type> <variable name>;-----
25 Integer age; // Variable Declaration
26 age = 76; -----
27
28 System.debug(name);
29 System.debug(age);
30
31 //<Class Name> <object Name> = new Constructor();
32 // new keyword will be used to allocate memory/location/place in Heap Memo
33 Animal.address = '7677734';-----
34 Animal dog = new Animal(); // GHJHGS
35 System.debug(dog.name); // GHJHGS ----
36 System.debug(Animal.address);-----
37
38 Animal cat = new Animal(); // GHJHGS
39 cat.name = 'Lucy'; //REGF -----
40 System.debug(cat.name);---
41 System.debug(Animal.address);

```

Assignment

Play around with Class & Instance Variable along with Access Modifiers

Full Video



Resources

- https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/langCon_apex_data_types.htm
- https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_classes_understanding.htm
- https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_classes_access_modifiers.htm