

# APEX - TRIGGERS

Apex triggers are like stored procedures which execute when a particular event occurs. It executes before and after an event occurs on record.

## Syntax

```
trigger triggerName on ObjectName (trigger_events) { Trigger_code_block }
```

## Executing the trigger

Below are the events on which we could fire trigger:

- insert
- update
- delete
- merge
- upsert
- undelete

## Trigger Example 1

Suppose we received a business requirement that we need to create an Invoice Record when Customer's 'Customer Status' field changes to Active from Inactive. For this, we will create a trigger on APEX\_Customer\_\_c object by following steps:

**Step 1:** Go to sObject

**Step 2:** Click on Customer

**Step 3:** Click on 'New' button in Trigger related list and add the trigger code as give below.

```
//Trigger Code
trigger Customer_After_Insert on APEX_Customer__c (after update) {
    List InvoiceList = new List();
    for (APEX_Customer__c objCustomer: Trigger.new) {
        if (objCustomer.APEX_Customer_Status__c == 'Active') {
            APEX_Invoice__c objInvoice = new APEX_Invoice__c();
            objInvoice.APEX_Status__c = 'Pending';
            InvoiceList.add(objInvoice);
        }
    }
    //DML to insert the Invoice List in SFDC
    insert InvoiceList;
}
```

## Explanation:

**Trigger.new:** This is the context variable which stores the records which are currently in context of trigger, either being inserted or updated. In this case, this variable has Customer object's records which has been updated.

There are other context variables which are available in the context: trigger.old, trigger.newMap, trigger.OldMap.

## Trigger Example 2

The above trigger will execute when there is update operation on Customer records. But suppose we wanted to insert invoice record only when Customer Status changes from Inactive to Active and

not every time.

For this, we could use another context variable `trigger.oldMap` which will store the key as record id and value as old record values.

```
//Modified Trigger Code
trigger Customer_After_Insert on APEX_Customer__c (after update) {
    List<apex_invoice__c> InvoiceList = new List<apex_invoice__c>();
    for (APEX_Customer__c objCustomer: Trigger.new) {
        //condition to check the old value and new value
        if (objCustomer.APEX_Customer_Status__c == 'Active' &&
trigger.oldMap.get(objCustomer.id).APEX_Customer_Status__c == 'Inactive') {
            APEX_Invoice__c objInvoice = new APEX_Invoice__c();
            objInvoice.APEX_Status__c = 'Pending';
            InvoiceList.add(objInvoice);
        }
    }

    //DML to insert the Invoice List in SFDC
    insert InvoiceList;
}
```

### Exaplnation:

We have used `Trigger.oldMap` variable which as explained earlier, is a context variable which stores the Id and old value of records which are being updated.