WHAT IS FRAGMENT ?

A Fragment represents a reusable portion of your app's UI. A fragment defines and manages its own layout, has its own lifecycle, and can handle its own input events.

Fragments cannot live on their own--they must be *hosted* by an activity or another fragment.

Fragments can be added, replaced, or removed.

You can manage it's back stack through it's host activity .

BENEFITS:-

MODULAR

NOT HEAVY AS ACTIVITY

REUSABLE

ADD FRAGMENT TO ACTIVITY:-

Add your fragment to the activity in 2 ways:-
In both cases You need to add a FragmentContainerView that defines the location where the fragment should be placed within the activity's view hierarchy

1.By defining the fragment in your activity's layout file.

<!-- res/layout/example_activity.xml -->

<androidx.fragment.app.FragmentContainerView

```
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/fragment_container_view"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:name="com.example.ExampleFragment" />
```

2.By defining a fragment container in your activity's layout file and then programmatically adding the fragment from within your activity.

```
<!-- res/layout/example_activity.xml -->
<androidx.fragment.app.FragmentContainerView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/fragment_container_view"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />
```

//the android:name attribute is missing here ,
Here we FragmentTransaction is used to instantiate a fragment and add it to the activity's layout.

Transaction like adding, removing and replacing fragments.
In your FragmentActivity, you can get an instance of the FragmentManager, which can be used to create a FragmentTransaction

OLD APPROACH :

```
 create fragment instance :
          val fragment : FragmentName =
FragmentName.newInstance()

          // for passing data to fragment
```

```
            val bundle = Bundle()
            bundle.putString("data_to_be_passed", DATA)
            fragment.arguments = bundle

val transaction = fragmentManager.beginTransaction()
transaction.add(R.id.LinearLayout1, firstFragment)
transaction.commit()
```

Example:-

```
//The fragment transaction is only created
when savedInstanceState is null\
New approach

class ExampleActivity :
AppCompatActivity(R.layout.example_activity) {
     override fun onCreate(savedInstanceState: Bundle?) {
       super.onCreate(savedInstanceState)
       if (savedInstanceState == null) {
          val bundle = bundleOf("some_int" to 0)
         supportFragmentManager.commit {
           setReorderingAllowed(true)
           add<ExampleFragment>(R.id.fragment_container_view,
args = bundle)
         }
       }
   }
}

// supportFragmentManager -Have access to
the FragmentManager through it

//FragmentManager.popBackStack(), the top-most fragment
transaction is popped off of the stack
```

Retrieving value from fragment

```
class ExampleFragment : Fragment(R.layout.example_fragment) {
    override fun onViewCreated(view: View, savedInstanceState:
Bundle?) {
        val someInt = requireArguments().getInt("some_int")
        ...
    }
}
```

You might never interact with FragmentManager directly if you're using the Jetpack Navigation library.

PERFORM A Transaction: Replace example

```
supportFragmentManager.commit {
    replace<ExampleFragment>(R.id.fragment_container)
    setReorderingAllowed(true)
    addToBackStack("name") // Calling addToBackStack() commits
the transaction to the back stack
}
```

```
//You can get an instance of FragmentTransaction from
the FragmentManager by calling beginTransaction(),
```

Very Imp Note:
If you don't call addToBackStack() when you perform a transaction that removes a fragment, then the removed fragment is destroyed when the transaction is committed, and the user cannot navigate back to it.
If you do call addToBackStack() when removing a fragment, then the fragment is only STOPPED and is later RESUMED when the user navigates back.

val fragment: ExampleFragment =

supportFragmentManager.findFragmentById(R.id.fragment_container) as ExampleFragment

or by tag , you can use

val fragment: ExampleFragment =
supportFragmentManager.findFragmentByTag("tag") as ExampleFragment

---

COMMUNICATION BETWEEN FRAGMENTS-

BY INTERFACE:

Make an Interface in your FragmentA
Implement the Interface of the FragmentA in your Activity
Call the Interface method from your Activity
In your Activity, call your FragmentB to do the required changes