Name Faizan Muhammad

Roll# Bsef18m023

A good link about git: <https://rubygarage.org/blog/most-basic-git-commands-with-examples#:~:text=Create%20a%20new%20file%20in,Repeat>.

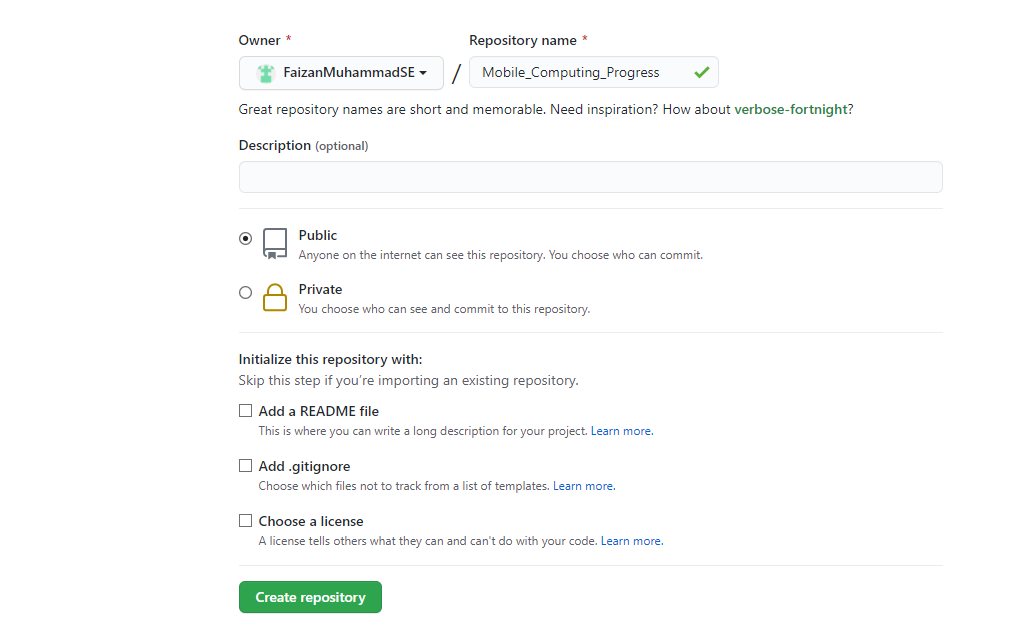
Version Control System is used to maintain record of updates done by you on your project, it is easily accessible from any place.

**Git is** a version control system that lets you manage and keep track of your source code history. **GitHub is** a cloud-based hosting service that lets you manage **Git** repositories.

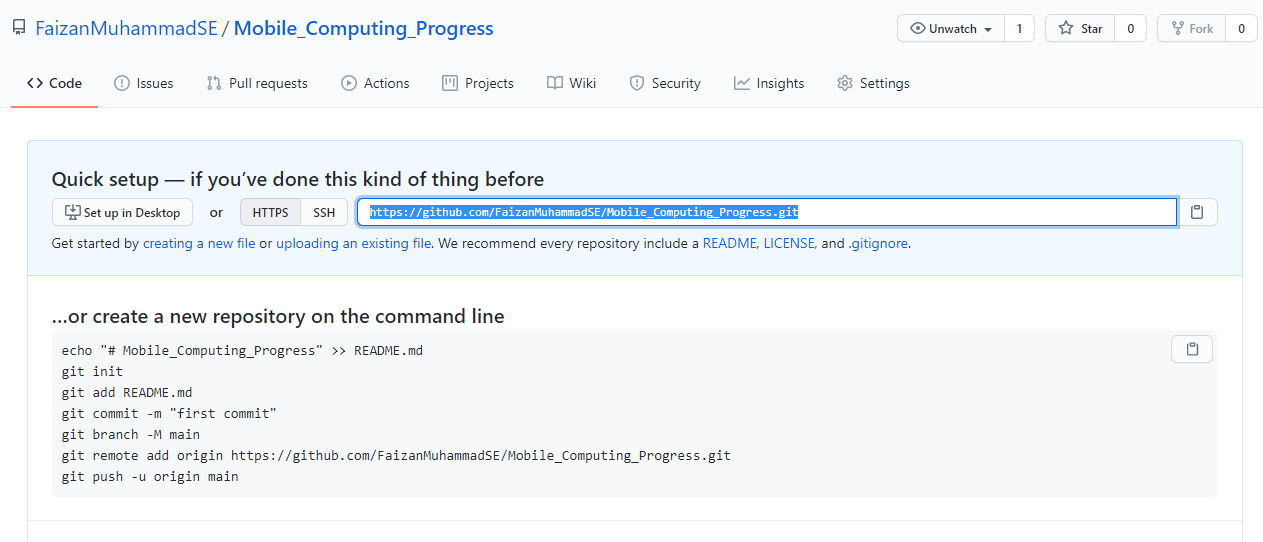
**GitHub:**

Here data is stored is the form of repository

**Public** repositories are available to everyone on internet.



The following link will be used to access repository, so you can give this link to concern people without any hesitation, they will interact with repository through this link



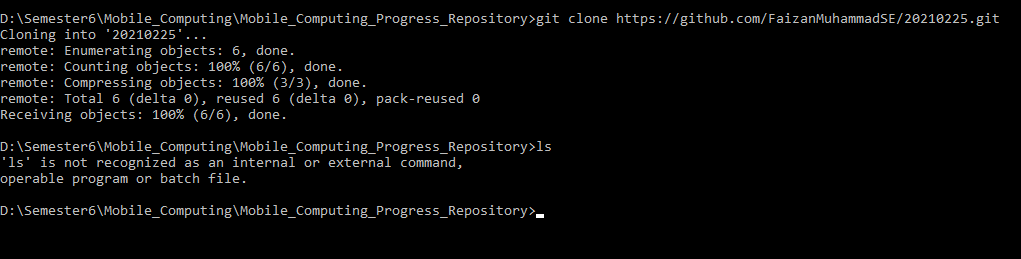
<https://github.com/FaizanMuhammadSE/Mobile_Computing_Progress.git>

**Clone of a repository:**

When you create a repository on GitHub, it exists as a remote repository. You can clone your repository to create a local copy on your computer and sync between the two locations.

When you clone a repository, you copy the repository to your computer

**Command: git clone url**



**Note: while cloning, your whole remote repository will be copied at current location(where you run command) and folder will have the same name as of repository.**

**Commad: git status**

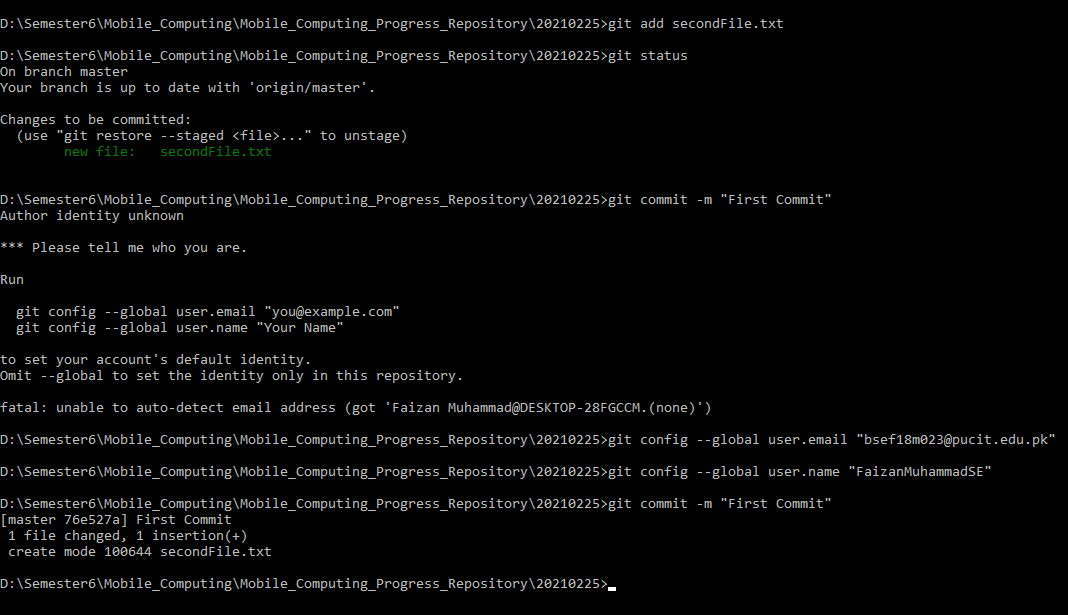
It will tells us about status of local repository and helps us to find out the files which are not added yet.

**Command: git add filename**

**git commit –m “message”**

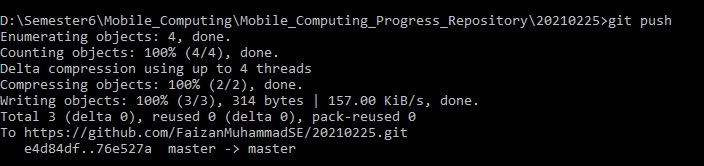
As you are working on your local computer and when you will do changings in your directory git will not be able to track it and know nothing about it. So you have to add it, by adding that specific file it will go in staging area(it is like a bucket of files which will go to remote repository GitHub), you can also remove files from staging area.

After adding, now files are in staging area, and now commit(attach message with it) it and now that file is in your local repository and git knows about it, and by commiting you get some ID, by using that you will know in future that what you did.

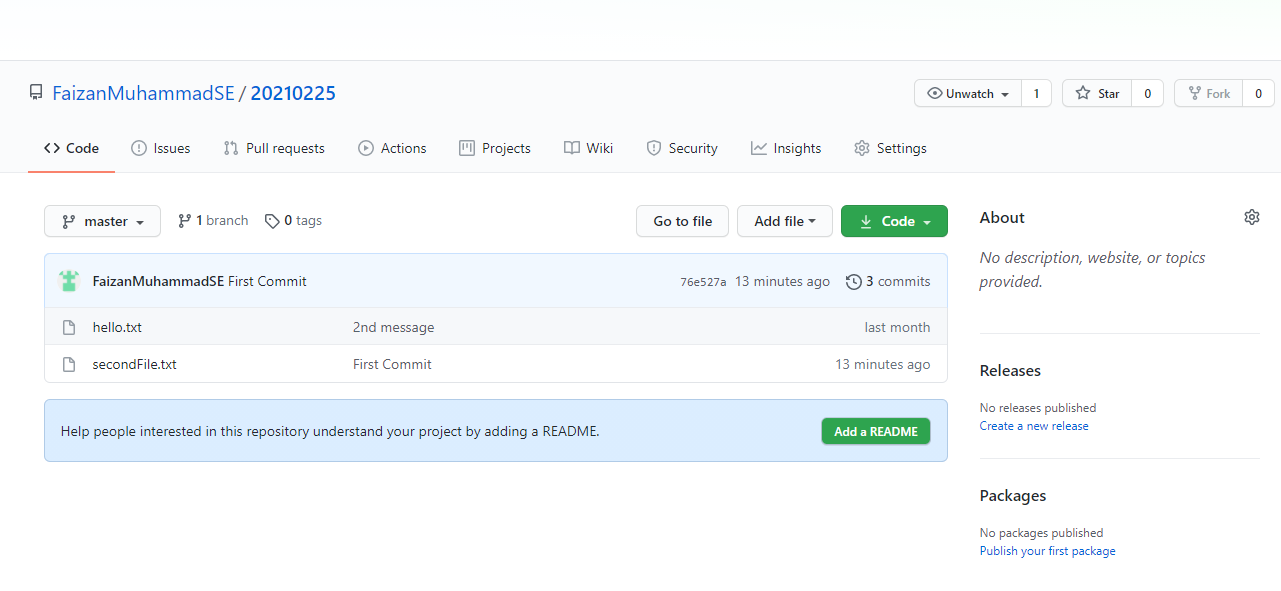


**Now we update our local repository but it is not synchronized with remote repository, so now push it to remote repository by using**

**Command: git push**

****

**Now both local and remote repository are synchronized/same**

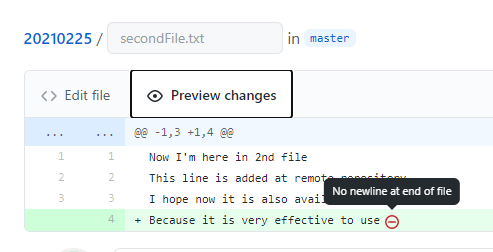
****

**So in short when you done any change in your project locally then use these 3 commands to synchronize your project with remote repository**

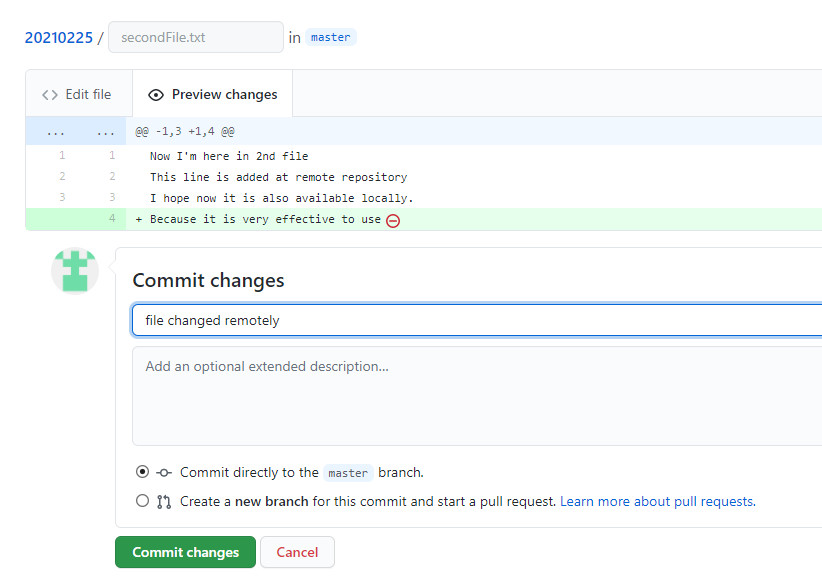
1. **git add filename Or git add . (for all files)**
2. **git commit –m “message”**
3. **git push**

even when you change a single line locally yoy have to performe these 3 commands to synchronize it

**How to synchronize project with local repo when you edit in at GitHub (local repo)**

**while editing file at GitHub you can also see changes done by you** ****

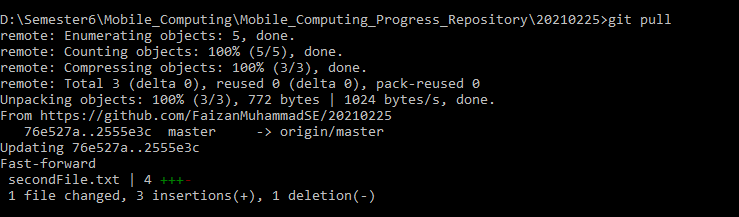
**When you edit file at GitHub, after editing, write commit message and click commit changes button present at bottom of file**

****

**Now it is changed remotely, how to synchronize with local repo**

**Command: git pull**

**By using this command your local repo will be refresh/reload and changes will be visible**

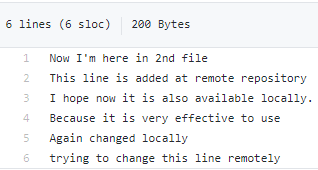
****

Merge Conflicts

It will arise when 2 user change the same line of code

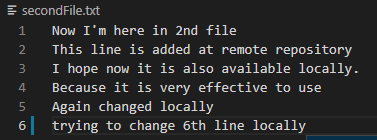
Lets understand it by example.

Suppose I change a specific line of a file at GitHub

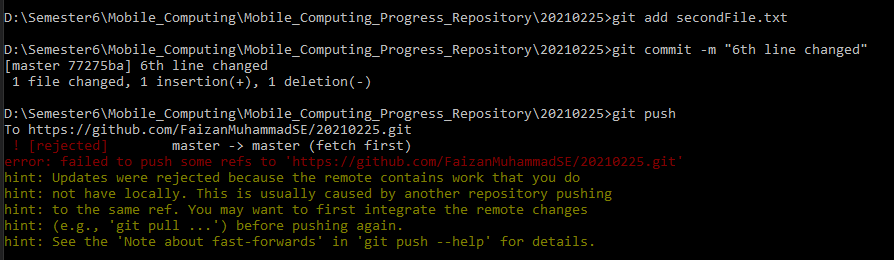


Here I changed 6th line remotely

Now I will change 6th line of same file at my local repo (computer)

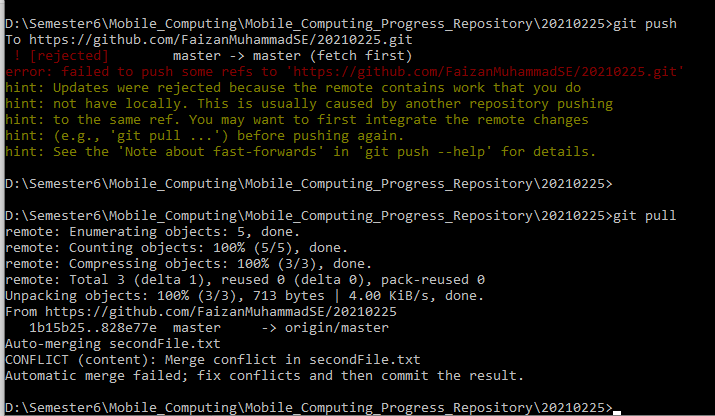


Now I want to sync this change at my remote repo, so will will use ‘git add command’



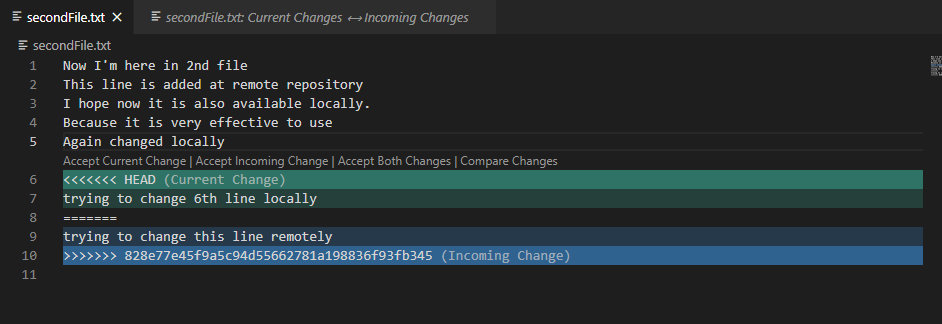
So giving error that, you have to pull the changed line, and changing it locally, so its giving suggestion that first pull it

After pulling from remote [git pull]



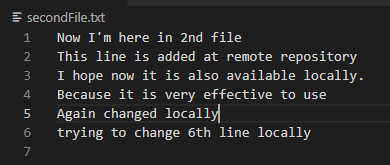
Inshort if you will push from your computer, it will show error

And when you pull it will show merge conflict error



And I select Accept Current Change

Now my local repo has



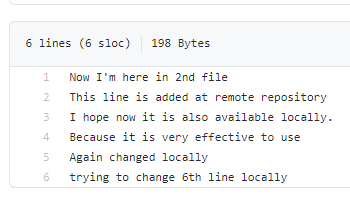
And I put it in staging area by using command

Git add secondFile.txt

Git commit –m “message”

Git push

Now remote repo is like



Note: merge conflicts will come when same line changed at remote and locally, and you trying to sync files, so first of all resolve conflit.

**To delete files, use command**

**git rm filename**

As you deleted files at local repo, but they are present at remote repo, so sync them

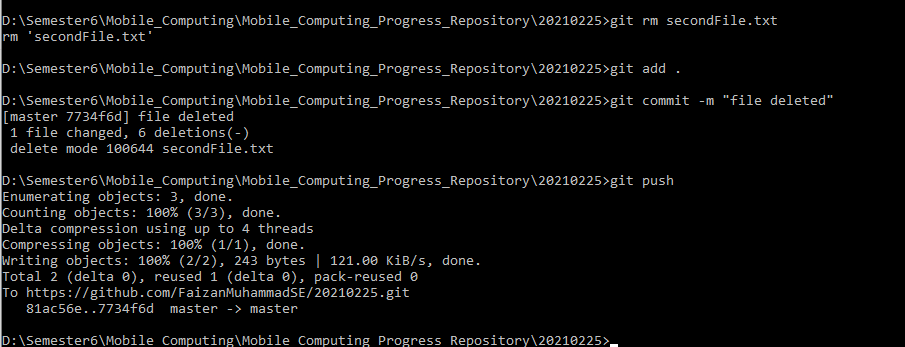
You again have to add all files in your working directory by using

git add .

git commit –m “message”

git push

Now deleted file will also be removed from remote repo



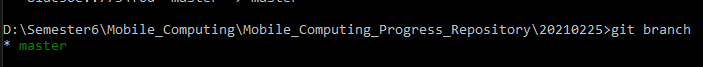
**Command: git log**

It will display us details of commits

Branching

We do braching when we want to do something with our project, but our current project is working fine, so we will make its branch and do changings in it, while new project will remain same as it was, until you merge your changings in it.

git branch ; will tell current branch ; root branch is master

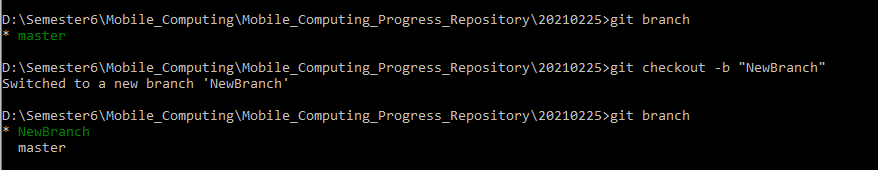


git create

git merge;

**To create new branch**

git checkout –b ”branchName” ;



Now we can do whatever we want in this

**To back from one branch to another use.**

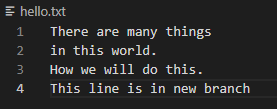
git checkout master ; it will lead us to master branch

git checkout NewBranch ; it will lead us to new branch

whenever we switch branch, our files and data will be changed

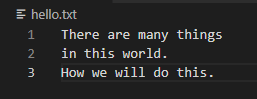
lets see by example

As now I’m in new branch, I’m going to add a 4th line in it, and commiting it, so new line is saved.



As I switched to master branch by running command ; git checkout master

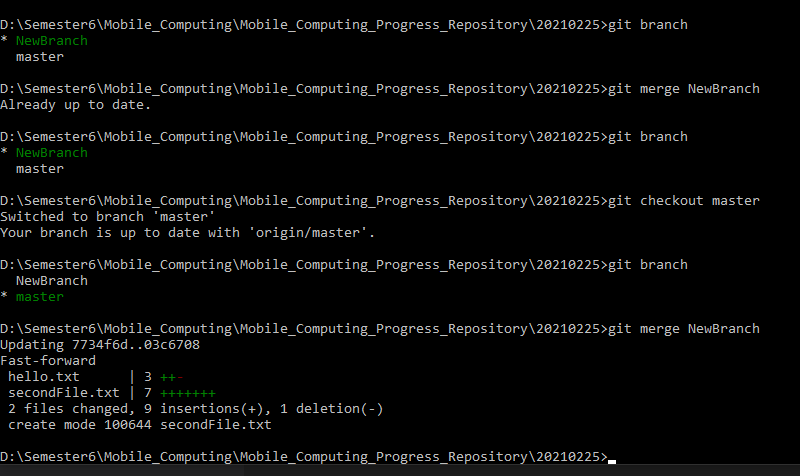
The file content change to previous(master) branch



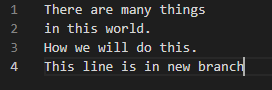
How to merge branch

Command: git merge NewBranch ; by running this now data of your newBranch is merged with master branch

Note: while running this command you should be at master branch

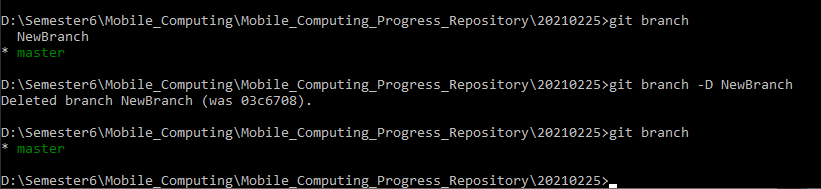


Data of file in master after merging



As we have merged with master branch, so if you want to delete NewBranch, then run this command

**Command: git branch –D NewBranch**

****

Now add,commit and push to also sync with remote repo, which you know already.

Lec#4

Intro to Android Visual Studio

* Package name should be uniquely define
* We will make user-interfaces in ‘MainActivity.java’
* In drawable we will place our images
* Gradle file will contain dependencies
* AVD stands for android virtual devices
* We will download devices from AVD to run our app on them
* We can also connect our phone with application
* Icons will be in minmap
* In values folder we will keep repeatedly used strings
* Logcat shows running things/files/processes
* In AndroidMenifest.xml we will have package name which will be globally uniquely identified
* MainActivity.java is the starting point our program
* Todo is good feature, it will tell us our pending work on opening project after closing

We can see full todo list under Todo tab

To make todo use this syntax //TODO: your statement will be here

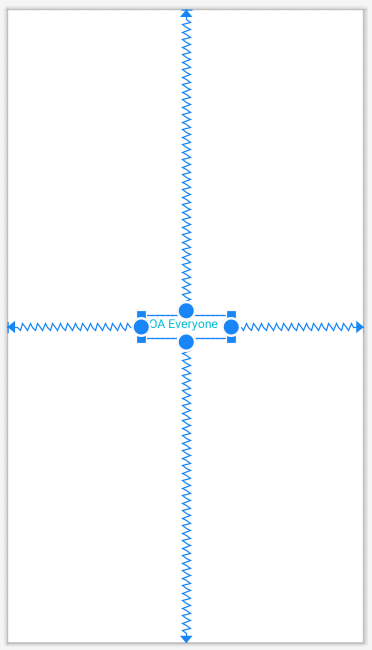
**There are few screens of program**

* **In this screen we add one textView.**
* **By using attribute text (search it in search bar ), we change text**
* **By using attribute textColor, we change its color**
* **Then run it**
* **The text is visible at top left corner, this is because we don’t define any contraint**

**First screen:**

****

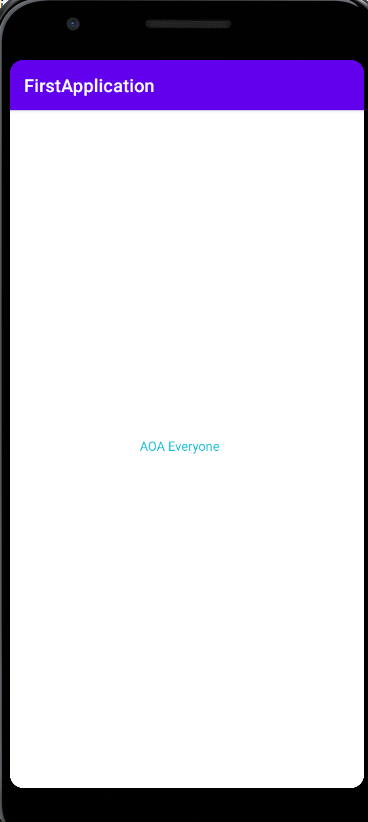
Now I define constraint for right, left, top and bottom



**Output after applying constraints**

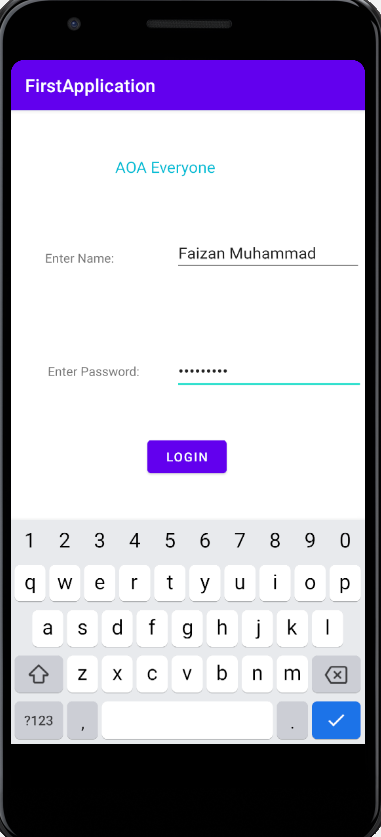
* **Output is same as we developed, it is because we defined constraints**

**2nd Screen:**

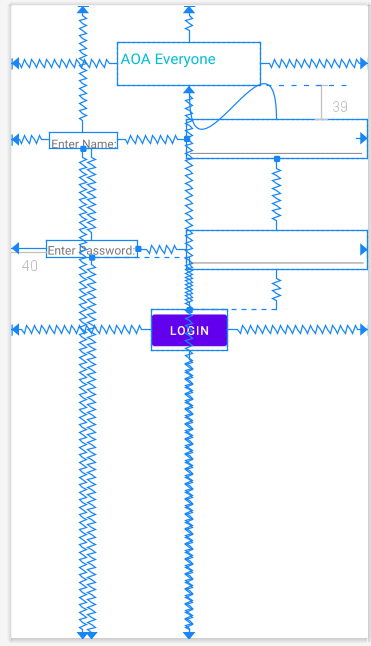
****

**3rd Screen:**

* First I made textView having text “AOA Everyone” and give it constraints
* Then I made a textView and a plain view to get user name, and I also gave them constraints
* Then I made another textView and passwordView to get user password, and also gave them constraints
* Then I made login button



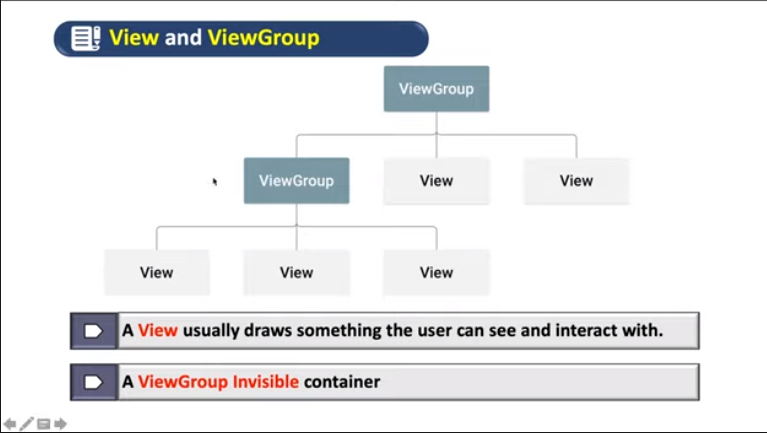
**Contrainsta are like this:**

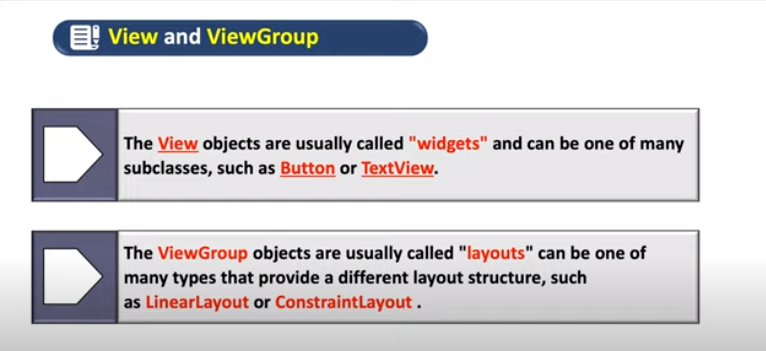
****

**Lec#5 Layout**

ViewGroup**: it is invisible container which has view or viewGroup**

View: **a view usually draw something which user can see and interact with, e.g button**





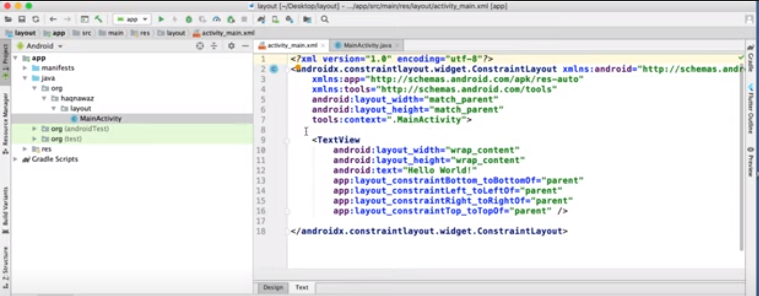
**Google recommended layout is constraint-layout**

**There will a root element, it could be view or view\_group, from which futher hirerarchy will extend**

**LinearLayout could be verticao** or **horizontal**

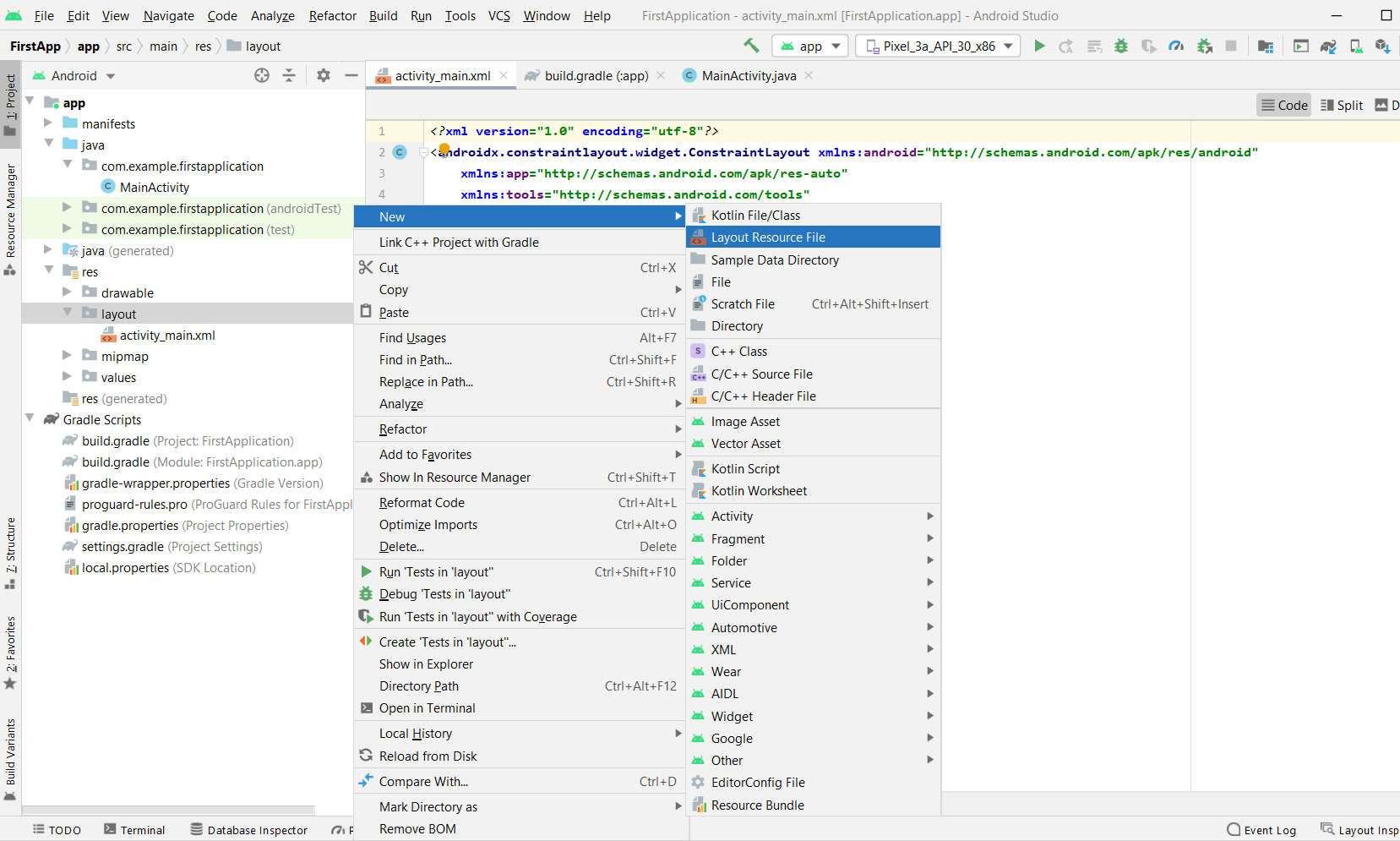


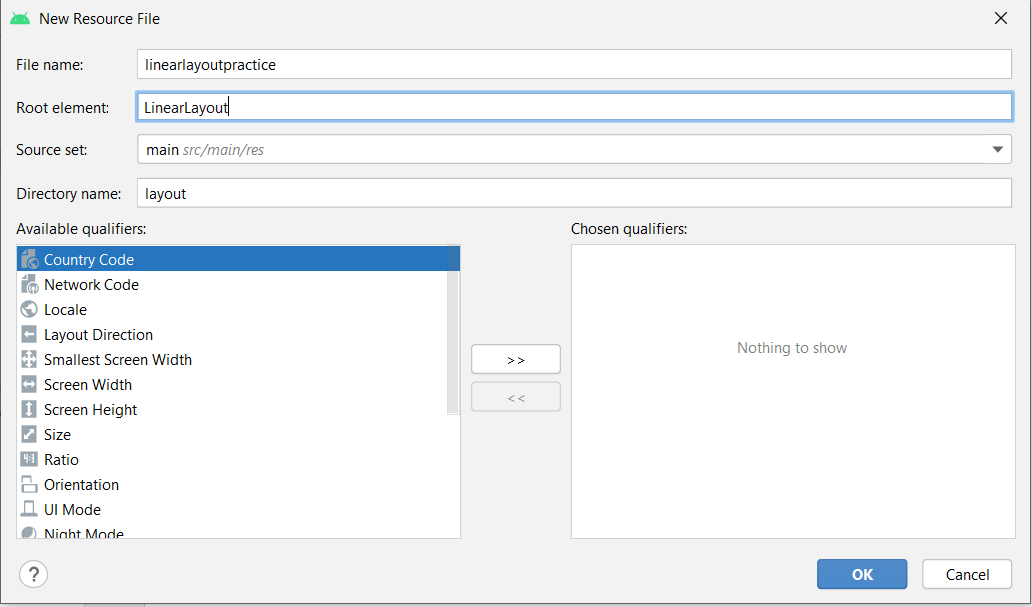
When we will start new activity, by default Constraint\_Layout used



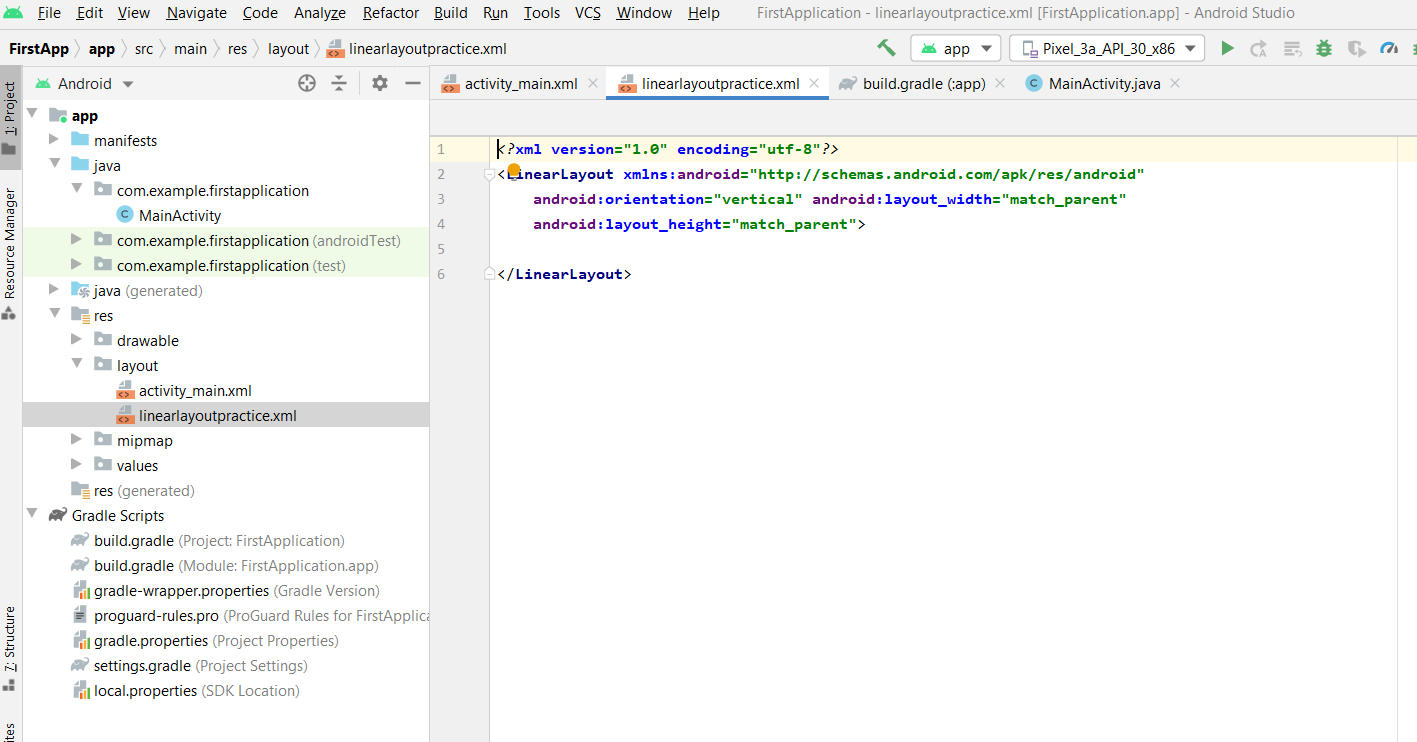
**We can change layout either of present .xml file or we can create new .xml file in resource or change its layout**

**How to create new .xml**

****

****

**Now new .xml file is created with your selected layout**

****

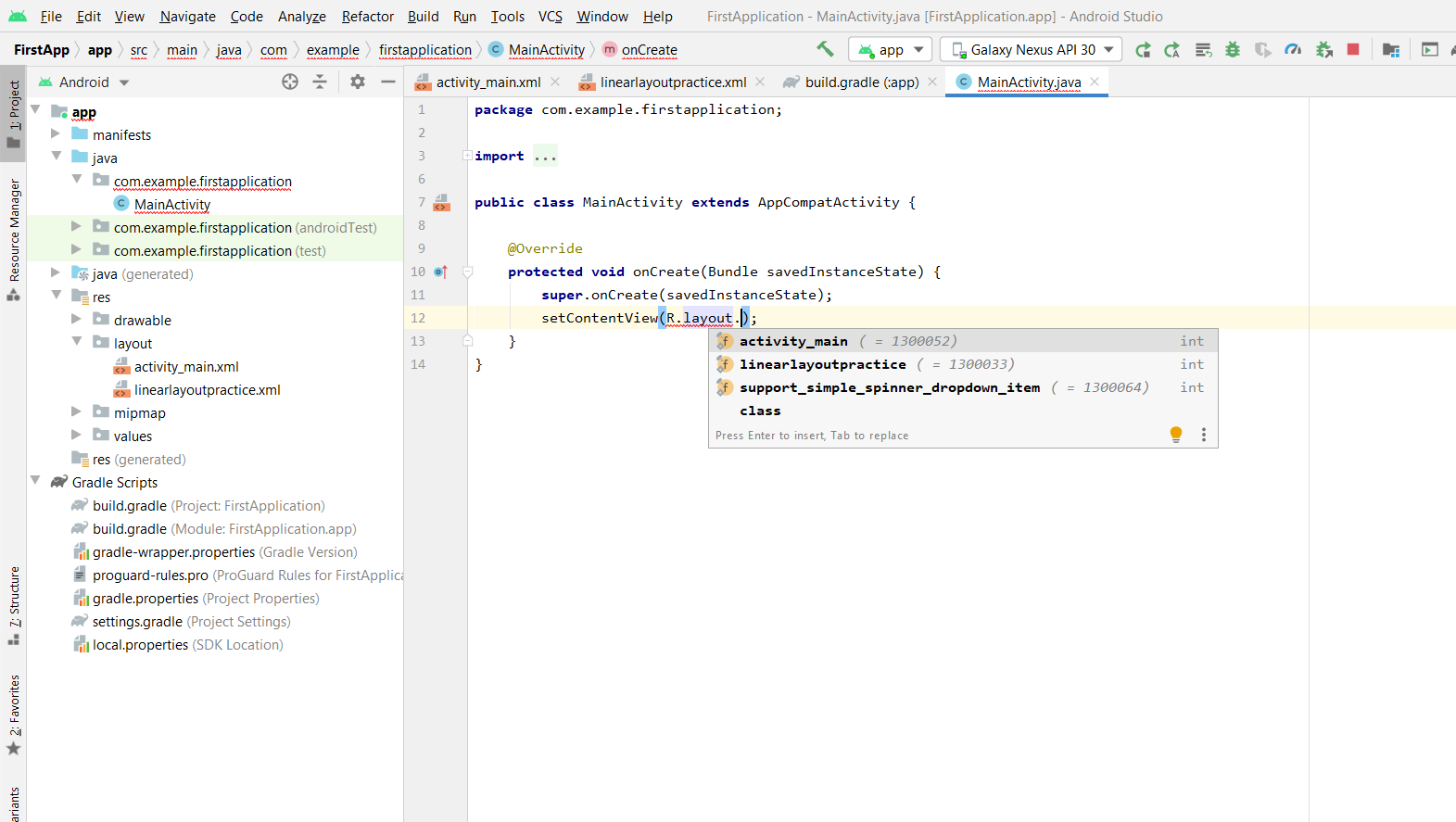
**By default orientation is vertical you can also do it horizontal**

****

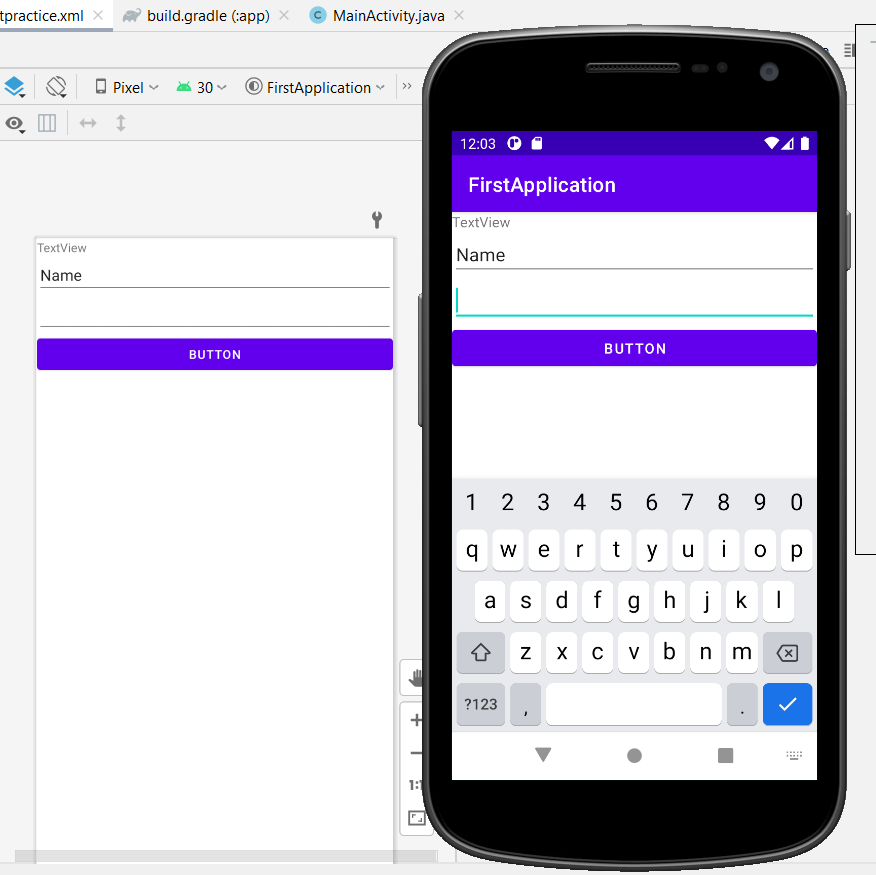
**As we have added new resource file now from where we should set that when program runs, data of our new resource/.xml file should be appear.**

**By default our setContentView is “activaty\_main.xml”**

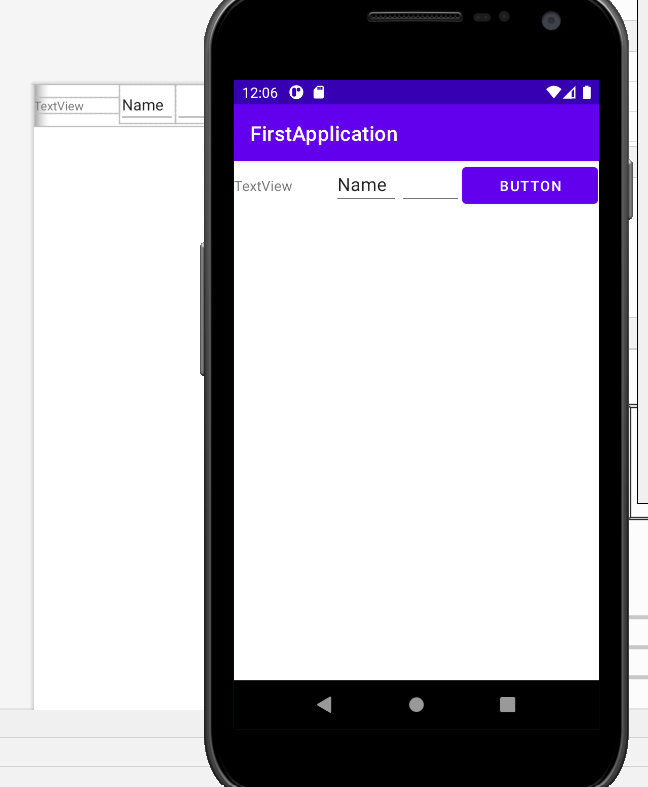
**As we know that program start from main, so lets go in main and change your resource/.xml file which you want**



Output using vertical mode of linear layout

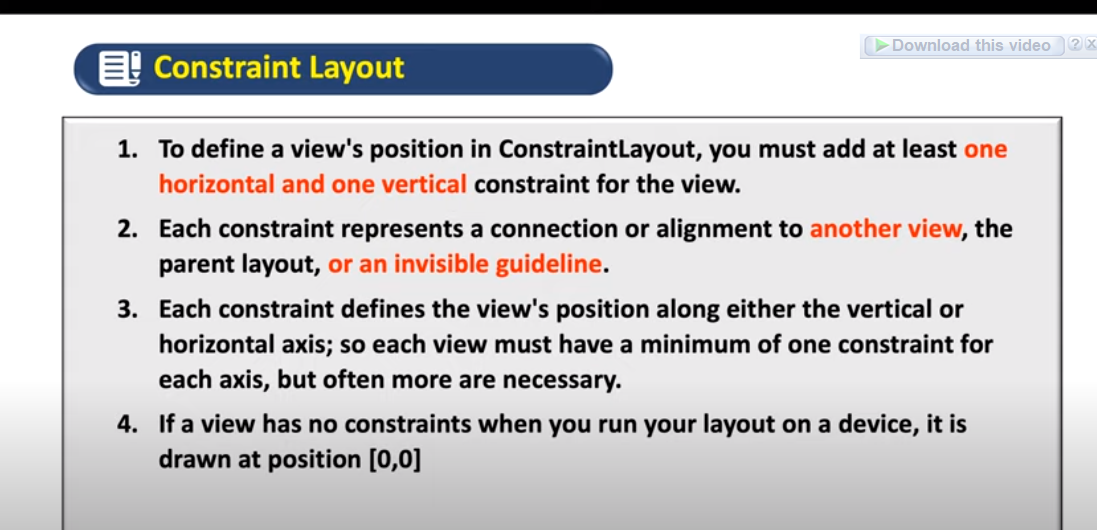


Output using horizontal mode of linear layout



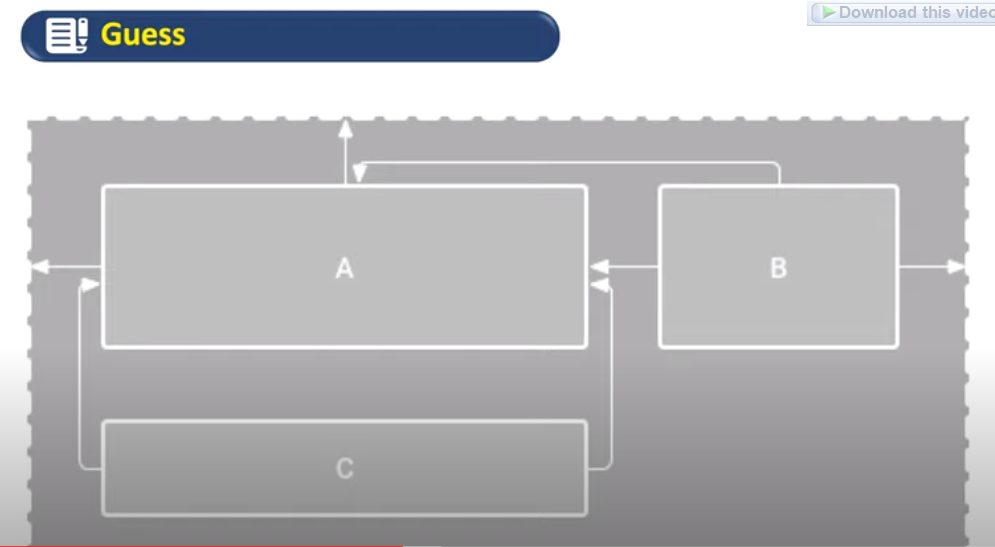
Layout Parameters

Every single [ViewGroup](https://developer.android.com/reference/android/view/ViewGroup.html) (e.g. [LinearLayout](https://developer.android.com/reference/android/widget/LinearLayout.html), [RelativeLayout](https://developer.android.com/reference/android/widget/RelativeLayout.html), [CoordinatorLayout](https://developer.android.com/reference/android/support/design/widget/CoordinatorLayout.html), etc.) needs to store information about its children's properties. About the way its children are being laid out in the ViewGroup. This information is stored in objects of a wrapper class ViewGroup.LayoutParams.



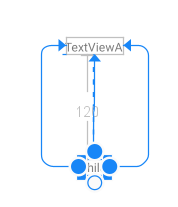
In this image ‘C’ has no constraint for y-axis

So it will stick at top, but its x-axis will be same



After exploring constraints I concluded that, movement of parent view will effect the movement of child view, while movement of child will not effect the parent view

Lets explore it by example

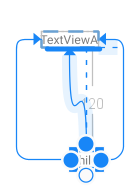


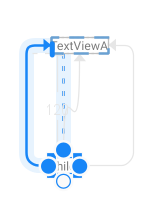
In this case child is constrained from top, right, and left side by TextViewA, so movement of child will not effect TextViewA

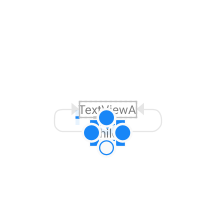
Here the question is this, at what positions child can move

Answer is child can’t go at right side, bcz its right side is constrained with TextViewA, the same is with left side, the same is with top

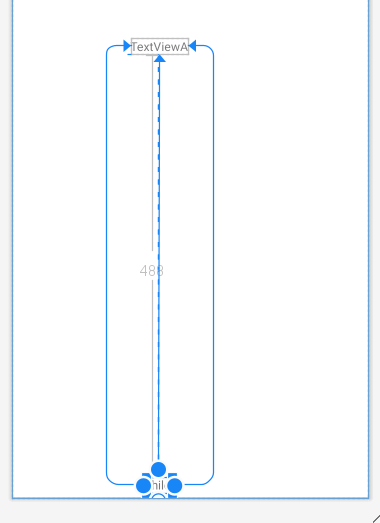
See



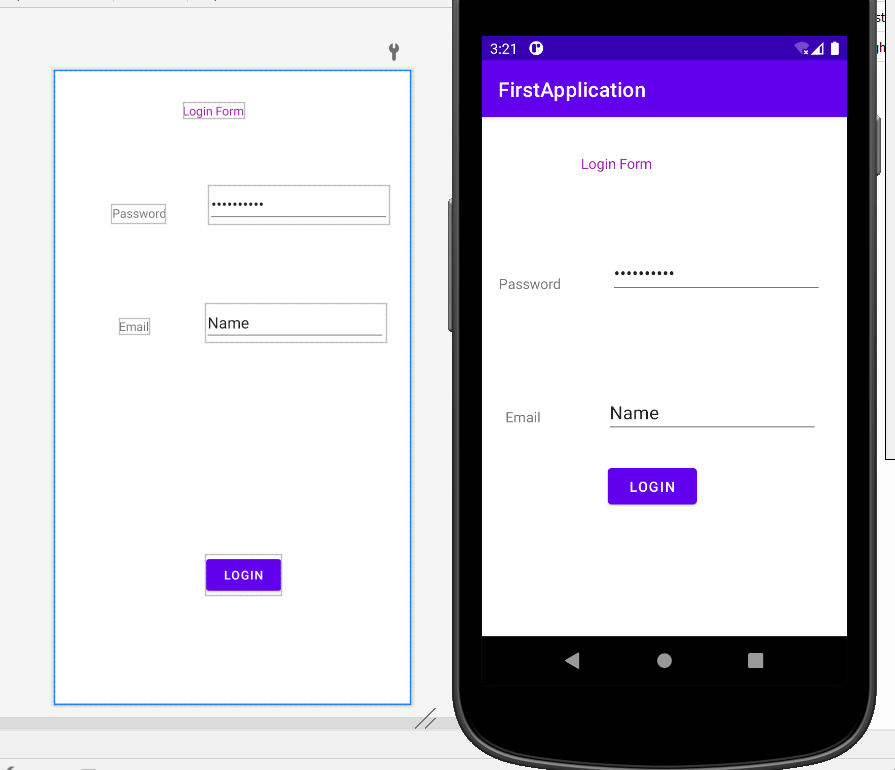




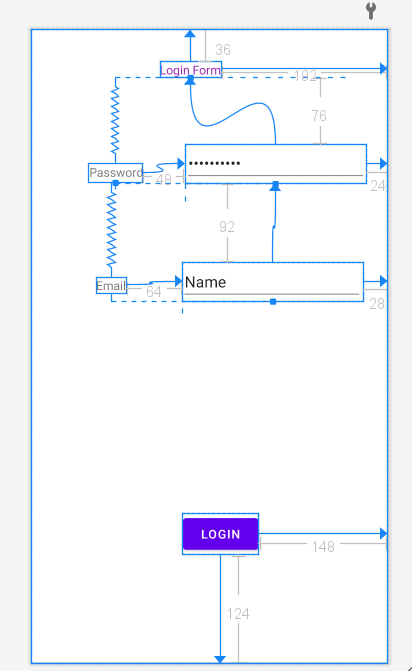
But as we know that there is no constraint at bottom so child can extend from bottom as you want



Lets make a login form

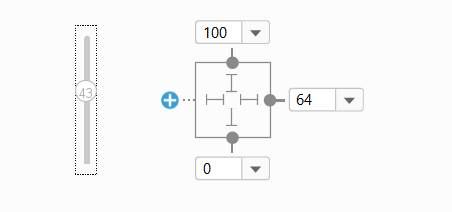


Its constraints are



To select all things/views for clearing constraint or other things use “Ctrl+A”

Here sliding bar will be used as a biasness of view , if horizontal bar then horizontal biasness, if vertical sliding bar then vertical biasness(current-case)



S