

## Git Concepts

### ❖ What is Git?

Git is a [distributed] Version Control System, which tracks the file changes

### ❖ What is GitHub?

GitHub is a cloud storage where we can store and manager project source code.

### ❖ What is Repository?

It is used for storing the software projects, it is most used for storing one project per repository.

### ❖ What is Branch?

It is a new or upgraded version of branch from which it is created.

**EX:** 1<sup>st</sup> version/branch of Flipkart project code that didn't had Flipkart groceries and the 2<sup>nd</sup> version/branch has the logic/code of Flipkart groceries but 2<sup>nd</sup> version doesn't mean it will not have the previous version's code, it is just an upgradation on top of previous version.

1. create branch: **git branch <new Branch Name>**.
2. Switch to another branch: a) **git switch <Branch Name>**. b) **git branch <Branch Name>**.
3. create and switch to new branch: **git checkout -b <new Branch Name>**.

### ❖ What is Commit?

It is a snapshot of a repo at a specific point in time.

Command: **git commit -m "commit message here"**

### ❖ What is Merging?

It is the process of bringing new changes from child repo to its parent repo

### ❖ What is Staging?

once all the changes are done and finalized, changes can be moved to staging by committing staging is the step before pushing the code to actual origin/target[remote repo].

### ❖ What is Stashing?

Stashing is nothing but to remove the local changes which are not committed yet and save it in a temporary place with a name called stash name.  
such stashed changes can be brought back by stash pop.

### ❖ What is Cloning?

Cloning is copying all the contents from the remote repository to a local repository.

### ❖ What are Merge conflicts?

When we try to merge branches but there are some changes which Git fails to identify and it needs our help to accomplish merging process.

**ex:** lets say we have 3 branches master, Foo, Bar and all are at sync and there are 2 dev's namely raven, xavier.

all branch's code:

```

1  using System;
2  public class Program
3  {
4      public static void Main()
5      {
6          Console.WriteLine('Hello World..!');
7      }
8  }

```

Now raven makes some changes to Foo:

```

1  using System;
2  public class Program
3  {
4      public static void Main()
5      {
6          Console.WriteLine('Hello World..! its xyz');
7      }
8  }

```

and this code[Foo] is now merged to master branch

and now master branch code looks exact like ravens branch 'Foo':

but now xavier makes some changes to branch 'Bar':

```

1  using System;
2  public class Program
3  {
4      public static void Main()
5      {
6          Console.WriteLine('Hello World..! its abc');
7      }
8  }

```

Now when xavier tries to merge branch 'Bar' with master it results in conflicts because git doesn't know whether to keep **xyz** change done by raven or **abc** done by xavier so it lets us to decide what to do and this is nothing but git conflict.