Branch

Tuesday 22 May 2018

16:10

What's a branch?

* In a collaborative environment, it is common for several developers to share and work on the same source code. Some developers will be fixing bugs while others would be implementing new features. Therefore, there has got to be a manageable way to maintain different versions of the same code base.

* This is where the branch functions comes to rescue. Branch allows each developer to branch out from the original code base and isolate their work from others. Another good thing about branch is that it helps Git to easily merge the versions later on.

1. What is a branch?

* Branch is essentially an independent line of development. You can take advantage of branch when working on new features of bug fixes as it helps to isolate your work from that of other team members.

History of 
released versions 
History of new 
feature 
develop ma. t 
History of bug 
fixing 

* Different branches can be merged into any one branch provided that they belong to the same repository.

* The diagram below illustrates how development can take place in parallel using branches.

* Branching enables you to isolate your work from others. Changes in the primary branch or other branches will not affect your branch, unless you decide to pull the latest changes from those branches.

* It is common practice to create a new branch for each task(bug fixing, new features etc.), which is a good practice because it allows others to easily identify what changes to expect, and also for backtracking purposes to understand why a particular code change is implemented.

Machine generated alternative text:
Add new feature 
Fix bug in video display 
Add a side bar 
Let's merge I 

1. Master branch

* Upon making the first commit in a repository, Git will automatically create a master branch by default. Subsequent commits will go under the master branch until you decide to create and switch over to another branch.

Machine generated alternative text:
master 
Commit 