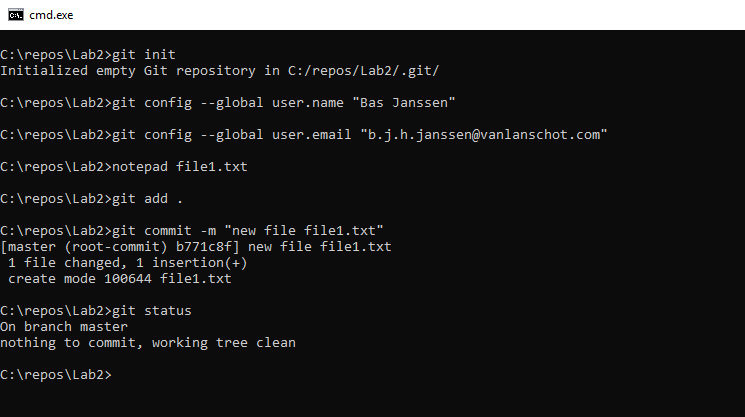
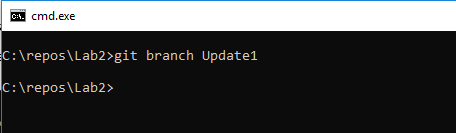
Lab2 : Branches

Voor deze lab en alle andere labs is het noodzakelijk de git client software geïnstalleerd te hebben.

* Maak en open een sub folder Lab2 aan in C:\repos en geef de volgende commando’s :
  + - git init
    - git config –global user.name “<voornaam> <achternaam>”
    - git config –global user.email "<email>"
    - notepad file1.txt ( voer een regel tekst in en save )
    - git add .
    - git commit -m “new file file1.txt”
    - git status

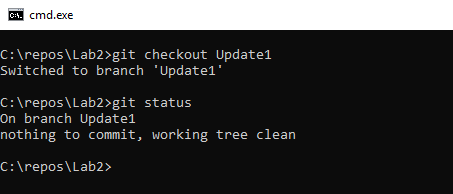


* + Het resultaat is een eerste commit met file1.txt in het snapshot
* Maak nu een **branch** aan met : git branch Update1 ( Update1 is de naam ) :

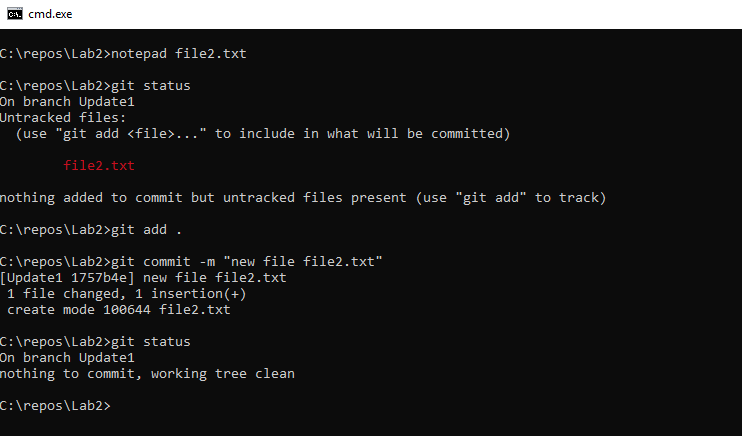


* + Geef eventueel : ‘git branch’ om de branches te tonen
* Checkout branch Update1 , geef : git checkout Update1

En check de status met : git status :



* + Herken dat je je in de branch Update1 bevind.
* Maak nu file2.txt
  + - open notepad, voer een regel tekst en sla op als file2.txt.
    - geef : git add .
    - geef : git commit -m “new file file2.txt”

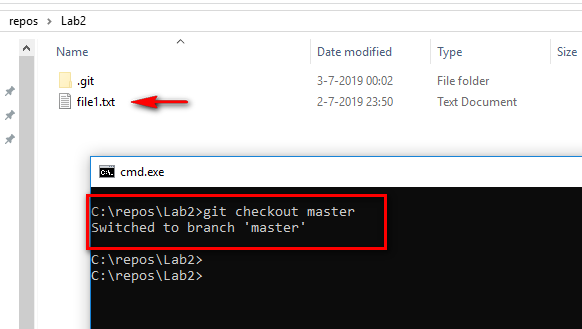


* + Status is nu dat we in branch Update1 en nieuwe file hebben toegevoegd.

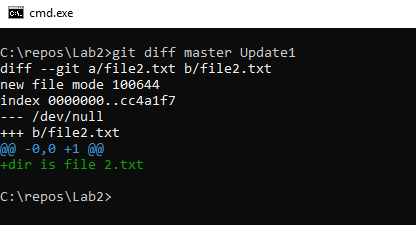
Dus de branch ‘Update1’ kent nu 2 files , file1.txt is meegekomen met het maken van de branch. Feitelijk maak je een kopie van de branch. En file2.txt hebben we net aangemaakt.

Branch ‘master’ kent alleen nog file1.txt

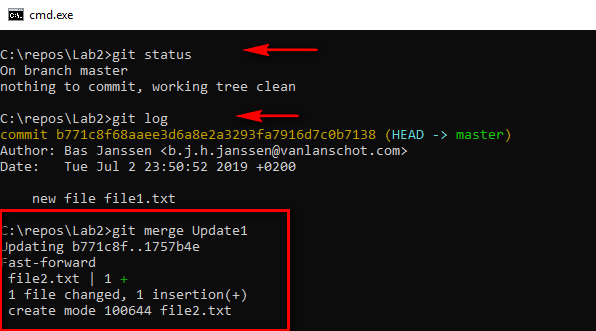
* Geef : ‘git checkout master’ en kijk in de windows explorer :



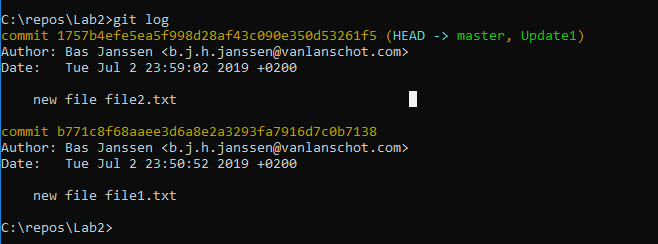
* + Je ziet nu alleen file1.txt omdat alleen deze file in de laatste commit van de master branch bekend is.
* Geef : git diff master Update1



* + Hier is het verschil tussen beide branches zichtbaar. Branch Update1 ( b ) heeft een extra file : file2.txt met de inhoud in het groen.
* Geeft nu : git merge Update1

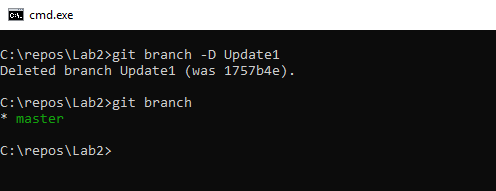


* + Het resultaat is dat in de master branch , de Update1 branch wordt samengevoegd met alle wijzigingen.



* + Het valt hier op dat HEAD nu zowel de laatste commit is van master als van Update1.
* We kunnen nu branch Update1 verwijderen omdat alle wijzigingen zijn doorgevoerd :

Dit doen we met : git branch -D Update1 :



* + Git branch : laat zien dat er nog maar 1 branch is : master
  + Met het juiste commando is het mogelijk de merge en de delete van de Update1 branch in één keer uit te voeren, maar dat is voor later …

Het resultaat is nu een folder Lab2 met twee files waarvan we file2.txt via een branch hebben toegevoegd aan de master branch.

Einde Lab2