

# 3 (!) Repositories make up a proper github workflow

1 Repository you want to contribute to at github (e.g. DF80E)

2 Repositories you need for your work ( 1 at github, forked, 1 at your pc to make the changes.)

Your repository at github is used to provide data to others. You will not work directly in this repository (mostly you will just push commits to it)

Let us go through it.

# GITHUB

DF80E uhsdr-git Repository

active-devel  
Branch  
@T0

master  
Branch  
@T0

FORK  
(clone)

MYREPO@github uhsdr-git  
Repository

active-devel  
Branch  
@T0

master  
Branch  
@T0

clone

## MY PC

MYREPO@pc uhsdr-git Repository

active-devel  
Branch  
@T0

master  
Branch  
@T0

# How a change is propagated into my (!) github Repository

Now the most simple workflow: Someone else creates a commit and this goes into the DF8OE repository active-devel branch. You cannot directly update your github repository, but you can do this in 2 steps.

T0 is the original state of the branches. T1, T2 etc. are commits done later

# GITHUB

DF80E uhsdr-git Repository

active-devel  
Branch  
@T0

master  
Branch  
@T0

MYREPO@github uhsdr-git  
Repository

active-devel  
Branch  
@T0

master  
Branch  
@T0

# Other PC (someone else)

REPO@other uhsdr-git Repository

active-devel  
Branch@T1

master  
Branch@T0

# MY PC

MYREPO@pc uhsdr-git Repository

active-devel  
Branch  
@T0

master  
Branch  
@T0

# GITHUB

DF80E uhsdr-git Repository

active-devel  
Branch  
@T1

master  
Branch  
@T0

MYREPO@github uhsdr-git  
Repository

active-devel  
Branch  
@T0

master  
Branch  
@T0

Merges  
Pull  
Request  
T1

Other (someone else)

REPO@other uhsdr-git Repository

active-devel  
Branch@T1

master  
Branch@T0

# MY PC

MYREPO@pc uhsdr-git Repository

active-devel  
Branch  
@T0

master  
Branch  
@T0

# GITHUB

DF80E uhsdr-git Repository

active-devel  
Branch  
@T1

Now active-devel branches at MYREPO@github  
And  
MYREPO@pc  
Are behind DF80E (and OTHER)

How to get these changes?

Merges  
T1

MYREPO@github uhsdr-git  
Repository

active-devel  
Branch  
@T0

master  
Branch  
@T0

Other

REPO@other uhsdr-git Repository

active-devel  
Branch@T1

master  
Branch@T0

MYREPO@pc uhsdr-git Repository

active-devel  
Branch  
@T0

master  
Branch  
@T0

# GITHUB

DF8OE uhsdr-git Repository

active-devel  
Branch  
@T1

master  
Branch  
@T0

MYREPO@github uhsdr-git  
Repository

active-devel  
Branch  
@T0

master  
Branch  
@T0

# Other PC (someone else)

REPO@other uhsdr-git Repository

active-devel  
Branch@T1

master  
Branch@T0

# MY PC

MYREPO@pc uhsdr-git Repository

active-devel  
Branch  
@T1

master  
Branch  
@T0

fetch df8oe  
+  
rebase dfe8oe/active-devel

# GITHUB

DF80E uhsdr-git Repository

active-devel  
Branch  
@T1

master  
Branch  
@T0

MYREPO@github uhsdr-git  
Repository

active-devel  
Branch  
@T1

master  
Branch  
@T0

## Other PC (someone else)

REPO@other uhsdr-git Repository

active-devel  
Branch@T1

master  
Branch@T0

## MY PC

MYREPO@  
Repository

active-devel  
Branch  
@T1

master  
Branch  
@T0

push  
origin



# How to make changes?

Before starting to make changes, make sure you have updated your working branch to the newest commit of df80e/active-devel. This will ease later integration and you'll have all the latest fixes and functions.

We assume now, that you did a successful rebase (which is a no-brainer, since you did not change anything yet). Your change's name is T2

# GITHUB

DF80E uhsdr-git Repository

active-devel  
Branch  
@T0

master  
Branch  
@T0

MYREPO@github uhsdr-git  
Repository

active-devel  
Branch  
@T0

master  
Branch  
@T0

## Other PC (someone else)

MYREPO@other uhsdr-git Repository

active-devel  
Branch@T1

master  
Branch@T0

## MY PC

MYREPO@pc uhsdr-git Repository

active-devel  
Branch  
@T2

master  
Branch  
@T0

# GITHUB

DF80E uhsdr-git Repository

active-devel  
Branch  
@T1

master  
Branch  
@T0

MYREPO@github uhsdr-git  
Repository

active-devel  
Branch  
@T0

master  
Branch  
@T0

Merges  
T1

Other (someone else)

REPO@other uhsdr-git Repository

active-devel  
Branch@T1

master  
Branch@T0

## MY PC

MYREPO@pc uhsdr-git Repository

active-devel  
Branch  
@T0

master  
Branch  
@T0

# GITHUB

DF80E uhsdr-git Repository

active-devel  
Branch  
@T0

master  
Branch  
@T0

MYREPO@github uhsdr-git  
Repository

active-devel  
Branch  
@T2

master  
Branch  
@T0

## Other PC (someone else)

REPO@other uhsdr-git Repository

active-devel  
Branch@T1

master  
Branch@T0

## MY PC

MYREPO@pc uhsdr-git Repository

active-devel  
Branch  
@T2

master  
Branch  
@T0

push  
origin

# GITHUB

DF80E uhsdr-git Repository

active-devel  
Branch  
@T2

master  
Branch  
@T0

DF80E Merges  
T2  
From your github repo

MYREPO@github uhsdr-git  
Repository

active-devel  
Branch  
@T2

master  
Branch  
@T0

## Other PC (someone else)

REPO@other uhsdr-git Repository

active-devel  
Branch@T1

master  
Branch@T0

## MY PC

MYREPO@pc uhsdr-git Repository

active-devel  
Branch  
@T2

master  
Branch  
@T0

# GITHUB

DF80E uhsdr-git Repository

active-devel  
Branch  
@T2

master  
Branch  
@T0

MYREPO@github uhsdr-git  
Repository

active-devel  
Branch  
@T2

master  
Branch  
@T0

## Other PC (someone else)

REPO@other uhsdr-git Repository

active-devel  
Branch@T1

master  
Branch@T0

## MY PC

MYREPO@pc uhsdr-git Repository

active-devel  
Branch  
@T2

master  
Branch  
@T0

# How to make changes while others have also made changes before my change was finished?

In this case rebase will help us to bring the external changes into your repository on the PC. If there are conflicts, because the other person changed in areas you touch as well, you'll have to resolve these conflicts. For now, assume the changes were made in different locations. T1 is the "other" change, T2 is still yours.

# GITHUB

DF80E uhsdr-git Repository

active-devel  
Branch  
@T0

master  
Branch  
@T0

MYREPO@github uhsdr-git  
Repository

active-devel  
Branch  
@T0

master  
Branch  
@T0

## Other PC (someone else)

MYREPO@other uhsdr-git Repository

active-devel  
Branch@T0

master  
Branch@T0

## MY PC

MYREPO@pc uhsdr-git Repository

active-devel  
Branch  
@T0

master  
Branch  
@T0



# GITHUB

DF80E uhsdr-git Repository

active-devel  
Branch  
@T0

master  
Branch  
@T0

Both you and the other guy  
did independent changes (T1  
and T2)  
You can't see the other  
changes, since they are still  
Only on the other pc

MYREPO@github uhsdr-git  
Repository

active-devel  
Branch  
@T0

master  
Branch  
@T0

## Other PC (someone else)

MYREPO@other uhsdr-git Repository

active-devel  
Branch@T1

master  
Branch@T0

## MY PC

MYREPO@pc uhsdr-git Repository

active-devel  
Branch  
@T2

master  
Branch  
@T0

# GITHUB

DF80E uhsdr-git Repository

active-devel  
Branch  
@T1

master  
Branch  
@T0

Merges  
T1

MYREPO@github uhsdr-git  
Repository

active-devel  
Branch  
@T0

master  
Branch  
@T0

Now the other guy makes a  
pull request and  
DF80E merges these  
changes into his github REPO

Other (someone else)

REPO (other uhsdr-git Repository)

active-devel  
Branch@T1

master  
Branch@T0

MY PC

MYREPO@pc uhsdr-git Repository

active-devel  
Branch  
@T2

master  
Branch  
@T0

# GITHUB

DF80E uhsdr-git Repository

active-devel  
Branch  
@T1

Now your local branch active-devel branches at  
MYREPO@github is behind 1 commit with  
respect to DF80E active-devel  
And  
MYREPO@pc is behind 1 commit and also ahead  
one commit (T2).  
How to get these changes?

MYREPO@github uhsdr-git  
Repository

active-devel  
Branch  
@T0

master  
Branch  
@T0

Other PC (some)

REPO@other uhsdr-git Repository

active-devel  
Branch@T1

master  
Branch@T0

MYREPO@pc uhsdr-git Repository

active-devel  
Branch  
@T2

master  
Branch  
@T0

# GITHUB

DF8OE uhsdr-git Repository

active-devel  
Branch  
@T1

master  
Branch  
@T0

The rebase operation will  
(possibly with your manual  
help) will merge the changes  
so that you have now both T1  
and T2 and the changes in  
order to make both work  
together

MYREPO@github uhsdr-git  
Repository

active-devel  
Branch  
@T0

master  
Branch  
@T0

## Other PC (someone else)

REPO@other uhsdr-git Repository

active-devel  
Branch@T1

master  
Branch@T0

fetch df8oe  
+  
rebase dfe8oe/active-devel

## MY PC

MYREPO@pc uhsdr-git Repository

active-devel  
Branch  
@T1+T2

master  
Branch  
@T0

# GITHUB

DF80E uhsdr-git Repository

active-devel  
Branch  
@T1

master  
Branch  
@T0

If you did not push your changes T2 to your github, a simple push origin will do the job, otherwise use "git push -f origin"

MYREPO@github uhsdr-git  
Repository

active-devel  
Branch  
@T1+T2

master  
Branch  
@T0

push  
origin

## Other PC (someone else)

REPO@other uhsdr-git Repository

active-devel  
Branch@T1

master  
Branch@T0

## MY PC

MYREPO@  
Repository

active-devel  
Branch  
@T1+T2

master  
Branch  
@T0

# GITHUB

DF80E uhsdr-git Repository

active-devel  
Branch  
@T1+T2

master  
Branch  
@T0

DF80E Merges  
@T1+T2  
From your github repo

MYREPO@github uhsdr-git  
Repository

active-devel  
Branch  
@T1+T2

master  
Branch  
@T0

## Other PC (someone else)

REPO@other uhsdr-git Repository

active-devel  
Branch@T1

master  
Branch@T0

## MY PC

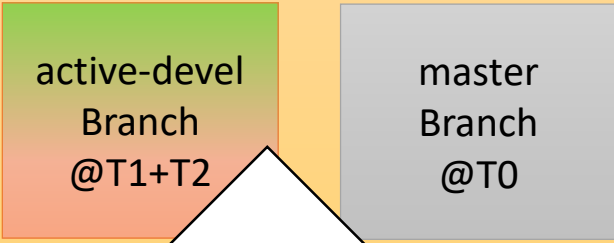
MYREPO@pc uhsdr-git Repository

active-devel  
Branch  
@T1+T2

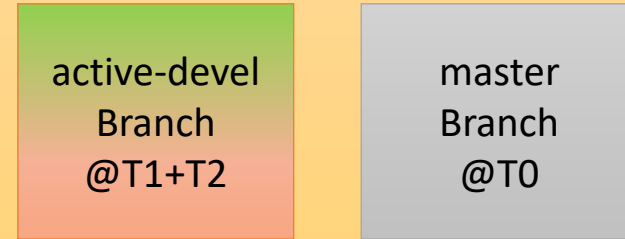
master  
Branch  
@T0

# GITHUB

DF80E uhsdr-git Repository

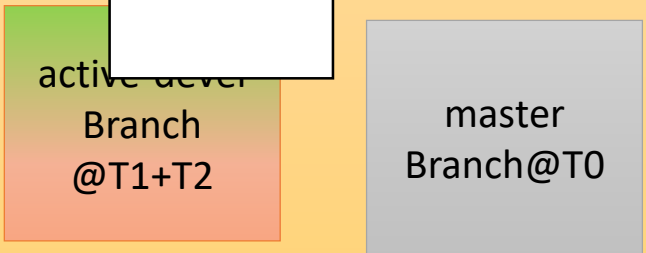


MYREPO@github uhsdr-git Repository



Other PC (one else)

REPO uhsdr-git Repository



Fetch and rebase

MY PC

MYREPO@pc uhsdr-git Repository

