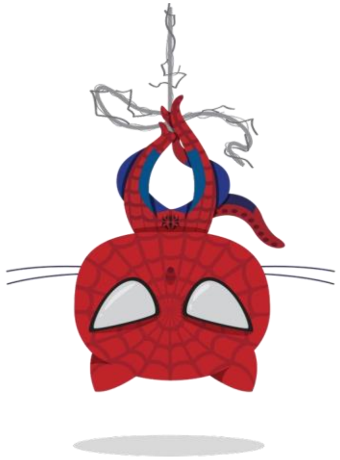
git

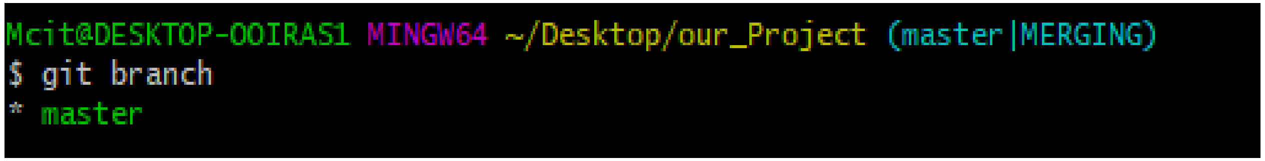
Day 2



Branching

If you need to work on a feature that will take some time, it's preferred to make a branch:

- To list all branches:
- $git branch



```
Mcit@DESKTOP-OOIRAS1 MINGW64 ~/Desktop/our_Project (master|MERGING)
$ git branch
* master
```

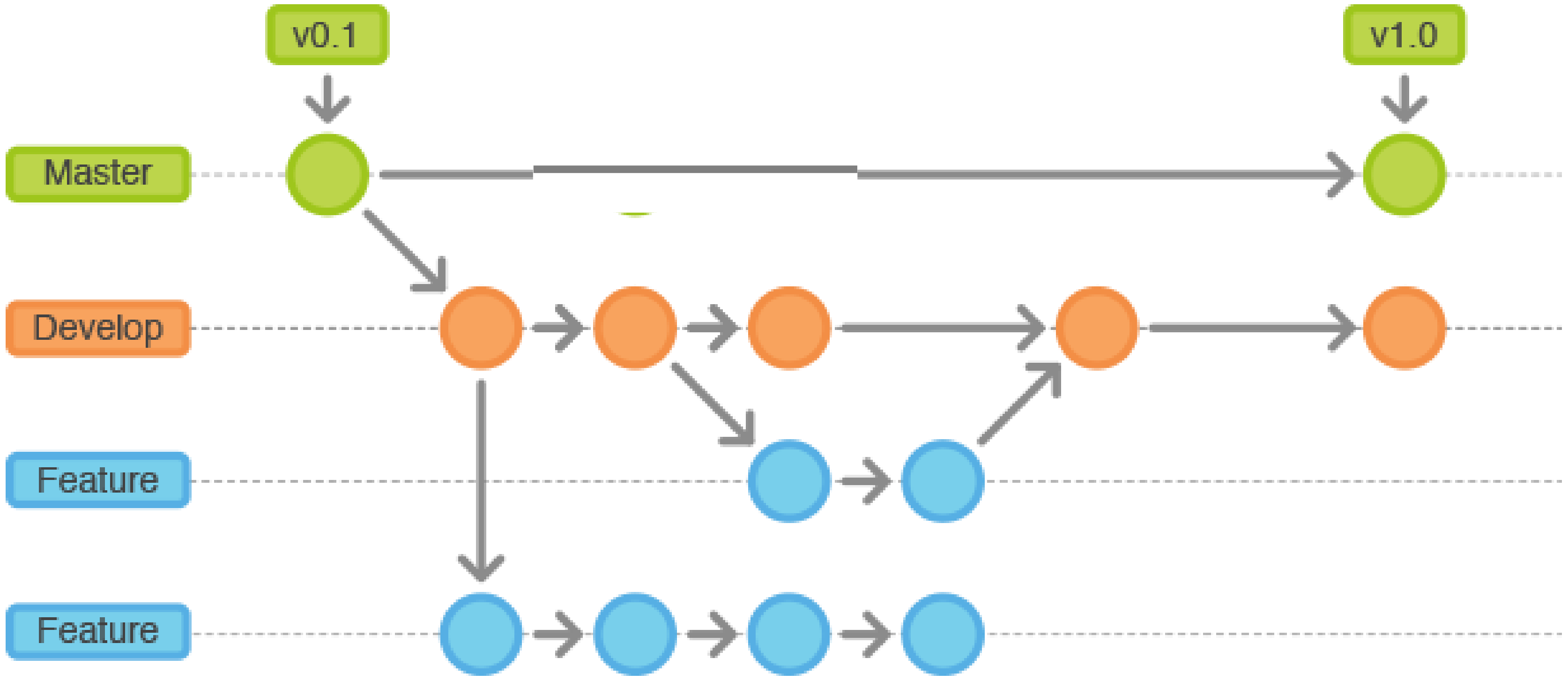
- To create new branch :

$ git branch my_branch

$git branch

master
my_branch

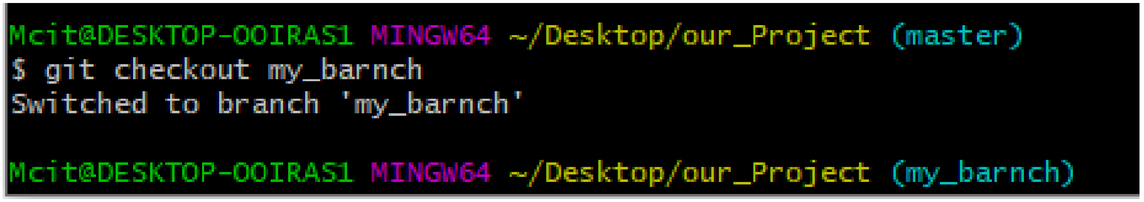
Branching



Switch to a branch

To switch to the branch :

```
$ git checkout my_branch
```



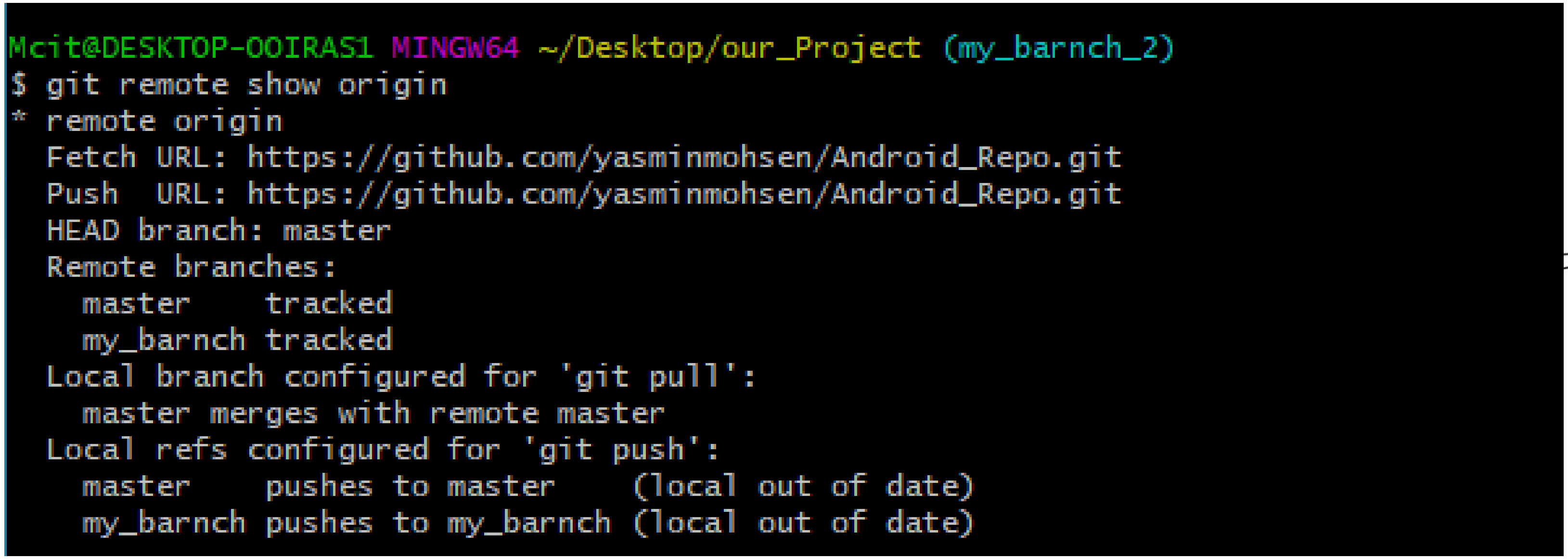
```
Mcit@DESKTOP-OOIRAS1 MINGW64 ~/Desktop/our_Project (master)
$ git checkout my_barnch
Switched to branch 'my_barnch'

Mcit@DESKTOP-OOIRAS1 MINGW64 ~/Desktop/our_Project (my_barnch)
```

- You can create new branch and switch to it in one step :

```
$ git checkout -b my_branch_2
```

Remote Show

$ git remote show origin



```
Mcit@DESKTOP-OOIRAS1 MINGW64 ~/Desktop/our_Project (my_barnch_2)
$ git remote show origin
* remote origin
  Fetch URL: https://github.com/yasminmohsen/Android_Repo.git
  Push  URL: https://github.com/yasminmohsen/Android_Repo.git
  HEAD branch: master
  Remote branches:
    master    tracked
    my_barnch tracked
  Local branch configured for 'git pull':
    master merges with remote master
  Local refs configured for 'git push':
    master    pushes to master    (local out of date)
    my_barnch pushes to my_barnch (local out of date)
```

Push to Remote Branch

It's preferable to pull all the changes from master in your branch before starting working in your branch to avoid conflicts when merging your branch with master:

```
$ git pull origin master
```

To push to the branch :

git <Remote Repo Name> <Branch Name>

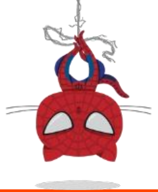
```
Ex: $ git push origin my_branch
```

Merging Branches

- **After finishing your work on the branch you have to merge it with master branch (assuming no changes were made to the master)**

```
$ git checkout master
$  git merge my_branch
$  git add .
$  git commit -m " your commit message“
$  git push origin master
```

Delete Branch



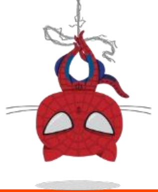
- When you’re done with a branch, you can safely remove it (this will not delete the branch if there are some files not merged with the master)

```
$ git checkout master
$ git branch –d my_branch (remove it locally )
$ git push origin --delete my_branch ( remove it remotely)
```

- this will remove the branch even if there are files not merged with the master

```
$ git checkout master
$ git branch –D my_branch
$ git push origin --delete my_branch
```

Tagging



A tag is a reference to a commit (used mostly for release versioning)

There are two types of tagging:

- **lightweight**: tag is very much like a branch that doesn't change – it's just a pointer to a specific commit.
- **Annotated tags**, however, are stored as full objects in the Git database. They're checksummed; contain the tagger name, email, and date; have a tagging message .

Create Tag

- To create a lightweight tag (This is basically the commit checks stored in a file – no other information is kept).

git tag<Tag_name>

Ex : $ git tag v0.0

- To show list of tags :

$ git tag

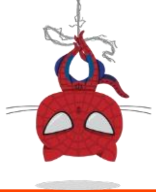
- To show specific tags :

$ git show v0.0

- To checkout to tag :

$ git checkout tags/v0.0 –b v0.0_branch

Create Tag / Push



- **To add a annotated tag**

```
$ git tag -a v0.0.1 –m “version v0.0.1”
$ git show v0.0.1
```

- **To push tags to remote repo :**

git push origin <tagname>

```
$git push origin v0.0
```

```
$git push --tags
```
* this will push all tags once

Tagging Later

- **You can also tag commits after you've moved past them.**

```
$git log
$git tag -a v0.0.1 9fceb02 –m "version v0.0.1"
```

It's the checksum number of the commit

Deleting Tags

- **To delete a remote tag:**

```
$git push origin –delete v0.0
```

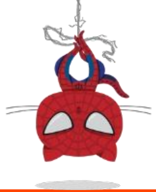
or

```
$ git push origin :refs/tags/v0.0
```

- **To delete a local tag:**

```
$ git tag –d v0.0
```

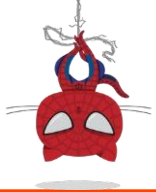
Stashing



- ❑ **Stash the changes in a dirty working directory away :**

- $git add
- $git stash
- $git stash save " your_messgae”
- $ git stash list
- $ git stash show
- $ git stash pop
- $ git stash pop stash@{id}   * id will equal to the id number of a specific stash file
- $ git stash apply
- $ git stash drop
- $ git stash clear

Stashing

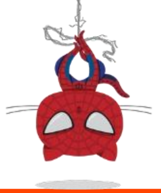


- ❑ **If you want to stash a specific staged file or change :**

- $git add
- $git stash -- filename

- $ git stash push -m “your msg" filename

History & Configuration



- To show the commits history :

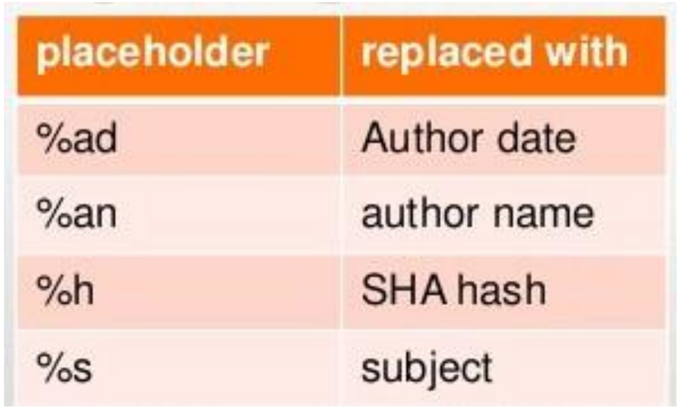
```
$ git log
```

- To show the history in one line :

```
$ git log --pretty=oneline
```

- To format the history log :

```
$ git log --pretty=format:"%h %ad- %s [%an]"
```

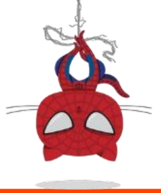


placeholder	replaced with
%ad	Author date
%an	author name
%h	SHA hash
%s	subject

Dates Ranges

- $ git log --until=1.minute.ago
- $ git log --since=1.day.ago
- $ git log --since=1.hour.ago
- $ git log --since=1.month.ago --until=2.weeks.ago

Ignoring Files



- Often, you'll have a class of files that you don't want Git to automatically add or even show you as being untracked.
- In such cases, you can create a file listing patterns to match ther named .gitignore.

```
$ touch .gitignore
```

Open .gitignore file and add the ignored files extensions

Ex :

```
*.log
```

README.MD

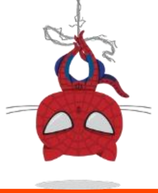
A README is a text file that introduces and explains a project. It contains information that is commonly required to understand what the project is about.

- Markdown is a simple way to format text that looks great on any device.
- It doesn’t do anything fancy like change the font size, color, or type — just the essentials, using keyboard symbols you already know.
- https://www.makeareadme.com/
- https://commonmark.org/help/

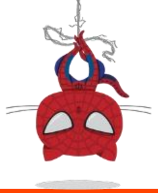
GitHub desktop application

https://desktop.github.com/

Issues



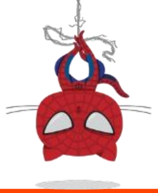
- https://docs.github.com/en/free-pro-team@latest/github/managing-your-work-on-github/creating-an-issue



Lab 2

- Create a new project on your local machine then push it to your remote repo.
- create two branches (dev and test) then create one file in dev branch(dev_file) and one file in test branch(test_file) then merge the div branch with main branch using gitbash, and merge test branch with main using pull request .
- Tell me how to remove them locally and remotly.

Lab 2



- create an annoted tag with tagname v1.4.
- push it to remote server.
- tell me how to list tags locally.
- tell me how to delete tag locally and remotely.
- Try to add new file and stash it