

IGNYTE254 & TECHNOSCIENCE

GIT Workshop



Hosting webpages on github

By Brian Wachira

IGNYTE254 & TECHNOSCIENCE

GIT Workshop



Hosting webpages on github

DATE : 30th March 2019, 0900-1200 hrs

VENUE : MS03

ENTRY: FREE

FOR MORE INFORMATION CONTACT +254707250844 / +254706245461



About ignyte254



We are a community interested in leveraging our tech skills with current existing technologies in the industry



ignyte254@gmail.com



@ignyte254

What We'll learn



- What git is. Why we need it. The different companies that offer git services
- How to use git
 - Installation
 - Setting up GitHub Account
 - Using Basic git commands (on terminal, windows cmd or git bash)
 - Using GitHub Desktop or gitKraken
- Branches and hosting webpages on github



So, what's Git



- It's a system that allows us to track our projects. I.e monitor changes & approve them. (A distributed version control system)
- By project I mean something that involves code
- Check out a more technical definition on [this link](#)



Why you need git



- 01 Ability to work remotely with anyone from any place
- 02 Backup your projects incase your machine fails
- 03 You can reference your projects to any potential client or recruiter

Companies that offer git services

- GitHub
- Gitlab
- Bitbucket
- Mercurial
- Find more on
<https://hackernoon.com/top-10-version-control-systems-4d314cf7adea>



How to use git



Installation

For **Windows users** download and install
the file from <https://git-scm.com/download/win>



For **linux and mac users**, Check out instructions in
<https://gist.github.com/derhuerst/1b15ff4652a867391f03>

Test your installation by running this command on any
command line

```
git version
```



Set Up A GitHub Account



What you need

- Your email (obviously).

Visit GitHub's [Official site](#) and sign up as a new user

and you're done.TM



Using git commands (cmd,terminal or gitbash applicable)

- **Creating a local Repository**

Create a folder. You can name it **gitworkshop**

Initialize a repository.

Repository - Is what Git uses to house your project locally (on your machine)

Enter your project folder (using a command-line) and type in this command to initialize your local repository

git init

You should get a success message like this

```
bmw@tdf:~/Documents/tech-communities/ignyte-254/gitworkshop$ git init
Initialized empty Git repository in /home/bmw/Documents/tech-communities/
ignyte-254/gitworkshop/.git/
```

● Tracking changes in your work

Use ***git status*** command to check the status of your repository

```
bmw@tdf:~/Documents/tech-communities/ignyte-254/gitworkshop$ git status
On branch master
  Your branch is up-to-date with 'origin/master'.
  nothing to commit, working tree clean
  Untracked files:
    (use "git add <file>..." to include in what will be committed)

      css/
      index.html
      js/
```



● Staging our changes for the repository

Use ***git add file_name.extension_type*** to stage our files for tracking.

You could also use the . operator to stage all the files at once.

```
bmw@tdf:~/Documents/tech-communities/ignyte-254/gitworkshop$ git add .
```

● Committing changes to the repository

Committing - Means adding our files & the changes made to our local repository.

Each include a message explaining what you are adding to your repo. The message should



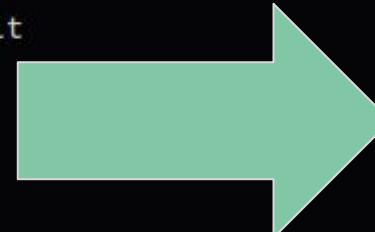
Each commit should include a message explaining what you are adding to your repo.

The message should be in present tense.
I.e It should complete the phrase “it will.....”

We use the command ***git commit -m “it will
<commit-message>”***

The **-m** stands for message.

```
bmw@tdf:~/Documents/tech-communities/ignyte-254/gitworkshop$ git commit -m "make initial commit"
[master (root-commit) 9f38e41] make initial commit
 6 files changed, 10391 insertions(+)
  create mode 100644 css/bootstrap.min.css
  create mode 100644 css/styles.css
  create mode 100644 index.html
  create mode 100644 js/bootstrap.min.js
  create mode 100644 js/jquery-3.3.1.js
  create mode 100644 js/scripts.js
```



Output message.
Created means
everything's ok



● Checking logs

Use ***git log*** to view your timeline.
I.e all the commits made, branches merged & who did them.

```
bmw@tdf:~/Documents/tech-communities/ignyte-254/gitworkshop$ git log
commit 9f38e410e1907f3019334062cbfe59f282a04928 (HEAD -> master)
Author: brianwachira <brianwachira7@gmail.com>
Date:   Fri Mar 22 15:07:11 2019 +0300

    make initial commit
```



More git knowledge?



Remote repositories (i.e pushing your code to Version Control Systems like github - [Right here](#)

Git docs - [Right here](#)

A more user-friendly resource - [Also here](#)



Working with Git using Graphical User Interfaces

[Github Desktop](#)

For Windows and Mac users

[Git kraken](#)

For Windows, Mac and Linux



GitHub Desktop

Download installer files on [this link](#)

Find resources on how to get started on [this link](#)



Git Kraken



Install files - Come with me & we'll go to
a place called [Candied Island](#)

Find resources on how to get started on
[this link](#)



Branches in git

- Working with them
- Github pages

What are branches?

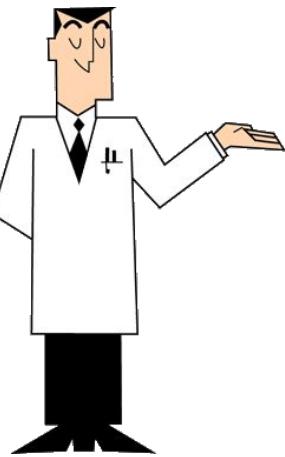


- Its a separate joint of your project
- Theres the main/default branch called Master.

Why We need Branches



- To run experiments
- To enable team work



Find more anti-batman
reasons [here](#)

Getting started with branches

- **Creating a new branch**

In cmd, We use the command `git branch branch_name` to create a branch.

```
bmw@tdf:~/Documents/tech-communities/ignyte-254/gitworkshop$ git branch  
container
```

To check existing branches, We use **`git branch`** command

```
bmw@tdf:~/Documents/tech-communities/ignyte-254/gitworkshop$  
git branch  
* container  
  master
```



Branches we have



Since we are in the main branch (master) We have to checkout to our new branch inorder work there. We use ***git checkout*** **branch_name** to move from one branch to another

```
bmw@tdf:~/Documents/tech-communities/ignyte-254/gitworkshop$ git checkout container
Switched to branch 'container' |  Output msg
```

NB : Commit all changes you make in your new branch. Otherwise they might be saved in the branch you have committed from

• Merging Branches



- It's basically joining a pair of branches together
- Make sure you have committed all code before merging
- There's something we call merge conflicts

Steps

1. Checkout to master branch
2. Use the command *git merge name_of_branch_you_wish_to_merge_with*

```
bmw@tdf:~/Documents/tech-communities/ignyte-254/gitworkshop$ git checkout master
Switched to branch 'master'  Step 1
bmw@tdf:~/Documents/tech-communities/ignyte-254/gitworkshop$ git merge container
Updating 9f38e41..fafcde3
Fast-forward
 index.html | 9 ++++++++
 1 file changed, 9 insertions(+)
```

 Step 2

You can delete a branch once you are done with it using
git branch -D name_of_branch_you_wish_to_delete

```
bmw@tdf:~/Documents/tech-communities/ignyte-254/gitworkshop$ git branc
h -D container
Deleted branch container (was fafcde3).  Output msg
```

- **Hosting with GitHub pages**



- **github:pages**
 - ★ Its a feature that allows us to host static webpages using a branch.
 - ★ You basically create a branch named **gh-pages** push it to your remote repository and that's it



Steps

- Add your remote origin and push your master branch.
- Create a branch named gh-pages
- Checkout to gh-pages branch
- Push it to your remote repository

```
bmw@tdf:~/Documents/tech-communities/ignyte-254/gitworkshop$ git remote  
add origin https://github.com/ignyte254/gitworkshop
```

Add origin

```
bmw@tdf:~/Documents/tech-communities/ignyte-254/gitworkshop$ git push -  
u origin master
```

Push to remote

```
Username for 'https://github.com': ^C
```

```
bmw@tdf:~/Documents/tech-communities/ignyte-254/gitworkshop$ git branch  
  gh-pages
```

Create branch

```
bmw@tdf:~/Documents/tech-communities/ignyte-254/gitworkshop$ git checko  
ut gh-pages
```

```
Switched to branch 'gh-pages'
```

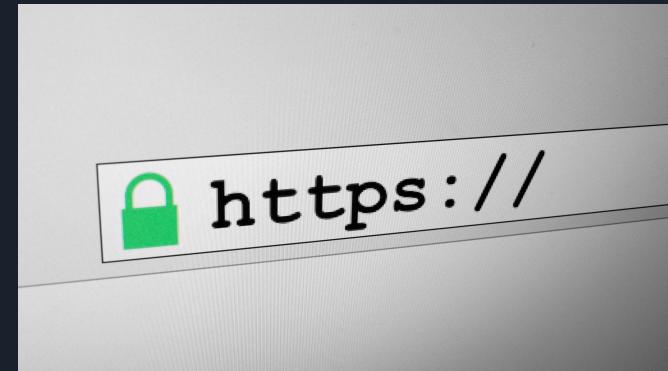
```
bmw@tdf:~/Documents/tech-communities/ignyte-254/gitworkshop$ git push -  
u origin gh-pages
```

Push to remote

```
Username for https://github.com: [REDACTED]  
Password for https://[REDACTED]: [REDACTED]
```

```
Counting objects: 3, done.  
Delta compression using up to 4 threads.  
Compressing objects: 100% (3/3), done.  
Writing objects: 100% (3/3), 493 bytes | 493.00 KiB/s, done.
```

So, where's the link??



- In your repository page, go to settings.
- Scroll to github pages. It's right there.

WB

That's all Folks!

"ANY QUESTIONS?"





About me

- **Android App Developer**
- **I can do Frontend too :)**
- **I play basketball for fun**



brianwachira7@gmail.com



<https://savoir-faire.netlify.com/>



@brianwachira



brianwachira



References?

Link to this slide