Starting with Git

Group Memebers

Hassan Mehmood BSCS-F19-M-67

Agenda

- What's a Version Control System?
- What is Git?
- Some Git commands
- What's about GitHub?
- Git in Action!

What's a Version Control System?

"An application that allows you to record changes to your codebase in a structured and controlled fashion."

Why do I need that?

- Makes it way easier to **undo errors** / **roll back** to earlier versions of code
- Makes it way easier to **share a codebase** between developers without creating conflicts
- Makes it way easier to **deploy changes** from development to staging or production environments

What is Git?

- Distributed Source Control system
- Open source, free (GNU GPL V2)
- Originally developed by Linus Torvalds for the development of the Linux Kernel in 2005
- Focus on speed and efficiency
- Quite a unique design and therefore sometimes a bit scary and difficult to understand

What is Git?

- Save snapshots, no differences
- Branching (lightweight & fast)
- Automatic merge of files
- Used on personal or very large projects, and for all size of teams

Git Commands

Getting and Creating projects

init

To create a git repository from an existing directory of files

```
$ git init
```

clone

If you want to get a copy of a project, you need to clone it

```
$ git clone [url]
```

Git Commands

Basic Snapshotting

git add

You have to add file contents to your staging area before you can commit them

```
$ git add index.php
```

git status

View the status of your files in the working directory and staging area

```
$ git status
```

Git Commands

Basic Snapshotting

git diff

Shows diff of what is staged and what is modified but unstaged

```
$ git diff
```

git commit

Records a snapshot of the staging area

```
$ git commit -m "My comment"
```

Git Commands

Basic Snapshotting

git reset

Undo changed and commits

```
$ git reset
```

git rm

Remove files from the staging area

```
$ git rm index.php
```

Git Commands

Branching and Merging

git branch

List, create and manage branches

```
$ git branch
$ git branch QA
```

git checkout

Switch to a new branch context

```
$ git checkout QA
$ git checkout -b live
```

Git Commands

Branching and Merging

git merge

Merge a branch context into your current one

```
$ git branch
$ git merge QA
```

git log

Show commit history of a branch

```
$ git log
```

Git Commands

Sharing and Updating Projects

git fetch

Download new branches and data from remote repository

```
$ git fetch
```

git pull

Fetch from a remote repo and try to merge into the current branch

```
$ git pull
```

Git Commands

Sharing and Updating Projects

git push

Push your new branches and data to a remote repository

```
$ git push
```

What's about GitHub?

- It’s a Git repository hosting service… but it adds many of its own features
- While Git is a command line tool, GitHub provides a web-based graphical interface
- It also provides access control and several collaboration features, such as wikis and basic task management tools

What's about GitHub?

- By default, all projects are public and free. In you want a private project, then pay
- You can clone any public repository, follow projects and developers, post comments, etc
- It’s becoming the Facebook’s for developers

Git in Action!

Go to your computer and start playing...

1. Create a new repository on GitHub
2. Clone this repository
3. Add new files
4. Commit and push them
5. Create a new branch and merge files