

# Git 101

Git is not Github



Link to slide: [goo.gl/u12TwU](https://goo.gl/u12TwU)

# What is Git?

Git most widely used modern version control system

Git is a mature, actively maintained open source project

Originally developed in 2005 by Linus Torvalds

Github is a git hosting provider like many other providers, eg bitbucket, gitlab ..

# What is Version Control

1. Version control systems are a category of software tools that help a software team manage changes to source code over time.
2. Version control software keeps track of every modification to the code in a special kind of database.

# The problem



```
<!DOCTYPE html>  
<html>  
<head>  
    <title></title>  
</head>  
<body>  
  
</body>  
</html>
```

new\_prject\_10\_01\_2018.html

# The problem



added\_paragraph\_and\_header\_\_10\_01\_2018.html

```
<!DOCTYPE html>
<!DOCTYPE html>
<html>
<head><title>Page Title</title></head>
<body>
    <h1>This is a Heading</h1>
    <p>This is a paragraph.</p>

    <h1>Another Heading</h1>
    <p>Another Paragraph</p>
</body>
</html>
```

# The problem



added\_form\_11\_01\_2018.html

```
<!DOCTYPE html>
<html>
<title>Page Title</title>
<body style="text-align: center;">
  <h1>This is a Heading</h1>
  <p>This is a paragraph.</p>
  <form method="post"
action="/action_page.php">
    First name:<br>
    <input type="text" name="name"
value="John Doe">
    <input type="submit"
value="Submit">
  </form>
</body>
</html>
```

# The problem



1.html



2.html



3.html



4.html



5.html



6.html



7.html



8.html



9.html



10.html

.....

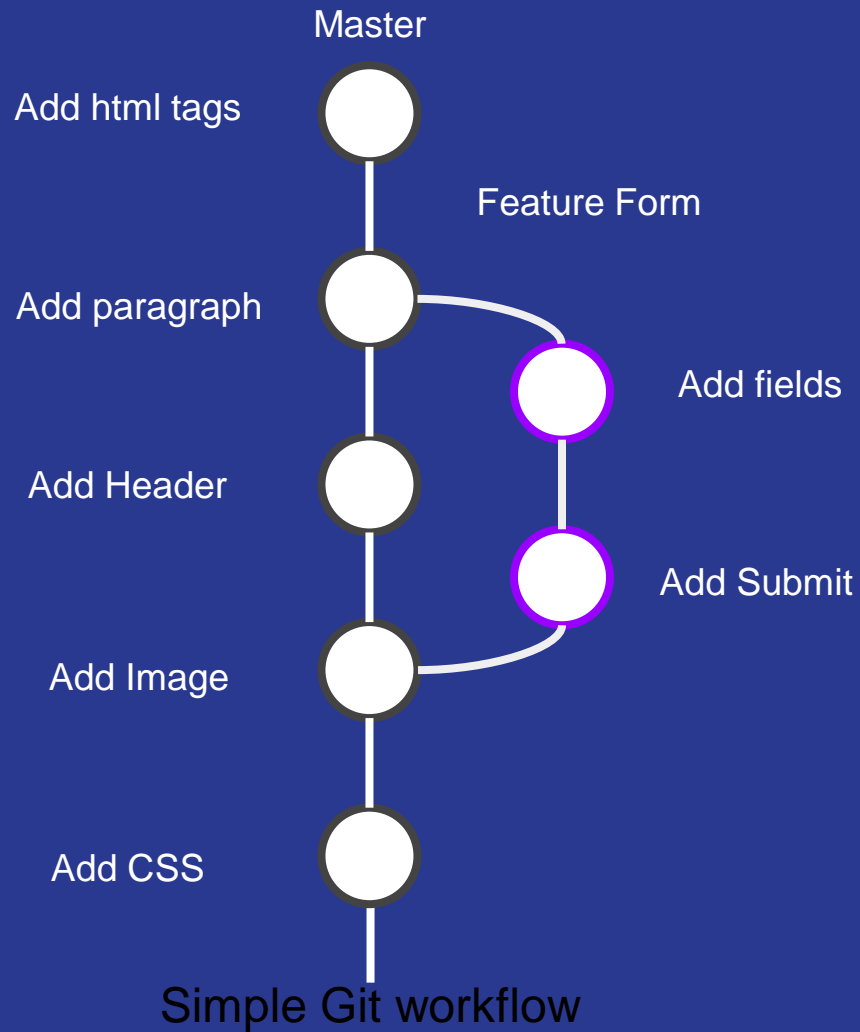
# The problem

1. Over the time system grows creating manual backup for every file will be difficult
2. It's not scalable to continue creating new file when new codes/features are added
3. Creating manual backups prone to errors
4. Difficult to find errors



# Git

is the solution



# Implementation

1. Create new Git repositories
2. Initiate git in existing Projects
3. Clone from a repository

# Step 1 (Create a project)

1. Create and Empty Project Folder
2. Open the folder
3. Click Right Mouse Button and select “Git Bash Here”
4. Type “**git status**” #expect an error not a git repo

## Step 2 (Initialize git and check status)

1. “**git init**” - initializing git in the project folder.
2. “**git status**” - check current status

## Step 3 (Create file and check status)

1. Create a sample file eg “index.html”
2. “**git status**” - check current status

# Step 4 (Edit the file)

1. Add basic html tags
2. “git status” - check current status

```
<!DOCTYPE html>
<html>
<head>
    <title></title>
</head>
<body>

</body>
</html>
```

# Step 5 (Add files to git index & first commit)

1. `git add .` # Add file contents to the index
2. `git status` - check current status
3. `git commit` -m "first commit basic html tags"

# Step 6 (Edit after commit)

1. Update the tags in “index.html”
2. “git status” - check current status

```
<!DOCTYPE html>
<!DOCTYPE html>
<html>
  <head>
    <title>Title</title>
  </head>
  <body>
    <h1>This is a Heading</h1>
    <p>This is a
paragraph.</p>

    <h1>Another Heading</h1>
    <p>Another Paragraph</p>
  </body>
</html>
```



# Step 7 (Create new file & commit with edited file)

1. Create a new file “page2.html” with basic html tags
2. “git status” - check current status
3. git commit -m “added page2 and updated index”

```
<!DOCTYPE html>
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
  </head>
  <body>
    <h1>This is a
Heading</h1>
    <p>This is a
paragraph.</p>

    <h1>Another
Heading</h1>
    <p>Another
Paragraph</p>
  </body>
</html>
```

# Step 8 (Add remote Repository)

1. Open gitlab and create a new project

`ProjectName`

1. Project path:

`https://gitlab.com/YourUserName/ProjectName`

1. Open the git window and type the “**git remote add origin https://gitlab.com/UserName/Projectname.git**”

`#Add the remote repository`

# Step 8 (Push to remote repository)

1. “**git push origin master**”
2. Will be Prompted for gitlab username & password
3. Codes are now pushed to remote repository
4. Checkthem in the Project / Activity / Push Events

# Step 9 (create new Branch & push the branch to remote)

1. “git branch” # show all branches
2. “git checkout -b new-branch-name source-branch”  
# create a new feature-add-form branch and switch to new branch
1. Edit page2.html and form tag from slide no 6
2. “git status”
3. “git add .”
4. “git commit -m “added form” “

# Step 9 (create new Branch & push the branch to remote)

7. “**git push -u origin name-of-current-branch**”  
# current branch is local only we are pushing it to remote server
7. Might be prompted for username and password if global user is not configured
8. You can see the new branch Repository / Branches

# Step 10 (Clone current project to a new folder)

1. Create a new folder
2. Navigate to the folder
3. Git Bash in to the folder
4. “**git clone repository path**” - cloning a repository  
Eg: `git clone https://gitlab.com/microBugTracker/test-project.git`
1. Will be prompted for user details for private projects  
“**git status**” - check current status  
#it's a private project

# Push (to cloned project)

1. Make some changes
2. “**git status**” - check current status
3. “**git add .**” - add all files
4. “**git commit - m “change message ” “** - commit message
5. “**git push -u origin branchname**” - push to remote server

# Push (to cloned project)

6. Then got to [gitlab.com/username/your-project](https://gitlab.com/username/your-project)
7. Create a merge request
8. And merge the branch if done



# Commands mostly used

1. `git status` # check current status
2. `git add .` # Add file contents to the index
3. `git commit -m "commit_message"` # Record changes to the repository
4. `git push origin branch-name` # Update remote refs along with associated objects
5. `git checkout branch-name` # Switch branches or restore working tree files

# In Depth Reference

1. Git best practices [link](#)
2. Basic git commands [link](#)
3. Git Docs [link](#)

# Git Providers



# Why Gitlab? (Github, Bitbucket, Gitlab comparison)

1. GitHub is free to use for public and open source projects.
2. Bitbucket is free for small teams (upto 5 users) unlimited private projects
3. For small teams, personal projects, or GitLab trials with unlimited time.



Q&A