Lesson Notes: Introduction to Git and Pushing Files to Repositories

Introduction to Git

Git is a version control system that allows developers to track changes in their code, collaborate with others, and maintain different versions of their projects. It is widely used in software development and DevOps workflows.

Setting Up Git

Before using Git, ensure it is installed on your system. You can verify the installation with:

```
git --version
```

If not installed, download and install it from https://git-scm.com/.

Configure Git

After installation, configure your user details:

```
git config --global user.name "Your Name"
git config --global user.email "your_email@example.com"
```

Pushing a File to a Public Repository

1. **Create a GitHub Repository:**

    - Go to GitHub and log in.
    - Click on **New Repository**.
    - Give it a name and set it to **Public**.
    - Click **Create Repository**.

2. **Initialize Git in Your Local Directory:**

    ```
    mkdir myproject
    cd myproject
    git init
    ```

3. **Add a File:**

    ```
    echo "Hello Git" > file.txt
    git add file.txt
    git commit -m "Initial commit"
    ```

4. **Connect to Remote Repository:**

git remote add origin https://github.com/your-username/repository-name.git

~~~
5. **Push to GitHub:**
```sh
git branch -M main
git push -u origin main
~~~

Pushing a File to a Private Repository

1. **Create a Private Repository:**
    - Follow the same steps as above but set the repository to **Private**.
2. **Authenticate Using Git Credentials:**
    - If prompted, authenticate using your **GitHub Username & Personal Access Token (PAT)**.
3. **Follow the Same Steps as Public Repo:**

git remote add origin https://github.com/your-username/private-repository.git git push -u origin main

~~~
If using SSH, first generate an SSH key:
```sh
ssh-keygen -t rsa -b 4096 -C "your_email@example.com"
~~~

Then add it to your GitHub SSH keys and use:

```
git remote add origin git@github.com:your-username/private-repository.git
```

Useful Git Commands

Command	Description
`git clone <repo_url>`	Clone a repository
`git status`	Check the status of the working directory
`git add <file>`	Stage changes to commit
`git commit -m "message"`	Commit changes with a message
`git log`	View commit history
`git pull origin main`	Pull the latest changes from remote
`git push origin main`	Push changes to remote
`git branch`	List branches
`git checkout -b new-branch`	Create and switch to a new branch
`git merge <branch>`	Merge a branch into the current branch
`git reset --hard <commit>`	Reset to a specific commit
`git stash`	Stash changes for later use
`git remote -v`	Show the remote repository URL

Conclusion

With these steps, you can initialize, track changes, commit, and push files to both public and private repositories. Practice regularly to master Git for efficient version control.