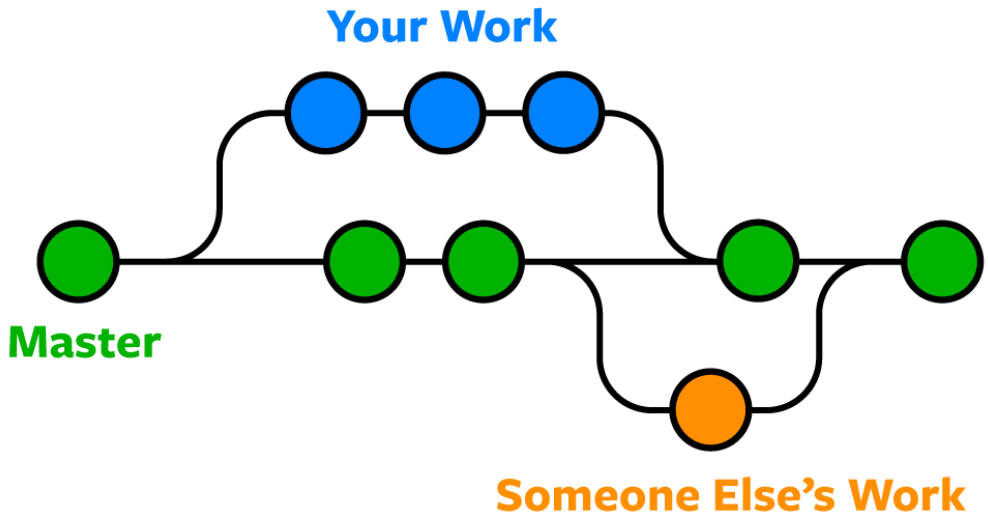
A Git branch is like a separate path in your project's history. It lets you work on different parts of your project without affecting the main code until you're ready



1. **Separate Paths**: Git branches allow for independent lines of development, enabling work on different features or fixes without affecting the main code.
2. **Isolation**: Each branch has its own commit history, changes, and modifications, ensuring that changes made on one branch do not impact others.
3. **Flexibility**: Branches can be easily created, switched between, merged, or deleted, providing flexibility in managing project development.
4. **Collaboration**: Branches facilitate collaborative development by allowing multiple team members to work on different tasks simultaneously.
5. **Experimentation**: Branches are commonly used for experimenting with new ideas or features without risking the stability of the main codebase.
6. **Visualization**: Git tools like `git log --graph` help visualize branch history, making it easier to understand the development flow and relationships between branches.
7. **Naming**: Descriptive branch names improve clarity and organization, helping developers understand the purpose or context of each branch.

Create new branch & confirm that is created & switching to new branch & push remote

```
git branch login-form
git branch
git checkout login-form
git push origin login-form
```

Merging branch to main

```
git merge login-form
git branch
```

Delete branch

```
git branch -d login-form
git branch
```

Delete branch from remote

```
git push origin --delete login-form
```