Short GitHub Tutorial

1 Introduction

GitHub is a web-based platform for version control using Git. It allows users to collaborate on projects and track changes in code over time. We will use it in this course to keep track of the exercises and solutions.

2 Creating a GitHub Profile

If you don't have one already, start by creating your GitHub profile at `https://github.com`. Your GitHub profile serves as your public coding portfolio.

3 Setting Up Git

Before using GitHub, install Git from `https://git-scm.com` and set your user info:

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

4 Authentication Options

You can authenticate with GitHub in two main ways:

4.1 SSH Keys (Recommended)

To avoid entering your password every time, set up an SSH key and add it to the SSH agent.

```
# Generate a new SSH key
ssh-keygen -t ed25519 -C "you@example.com"

# Start the ssh-agent in the background
eval "$(ssh-agent -s)"

# Add your SSH private key to the agent
ssh-add ~/.ssh/id_ed25519
```

Copy the public key to GitHub under `Settings > SSH and GPG keys`.

4.2 HTTPS with a Password Manager

Alternatively, you can use HTTPS URLs for Git and store your credentials using a helper like Git Credential Manager or pass.

```
# Example of cloning with HTTPS
git clone https://github.com/username/repo-name.git

# When prompted, enter your GitHub username and personal access token (PAT)
# instead of password
```

This method is simpler to set up but requires entering a token once or storing it securely.

5 Creating a New Repository

Go to `https://github.com`, log in, and click `New Repository`. Name it and choose options like `Initialize with README`.

6 Cloning a Repository

Clone a remote repo to your local machine (using SSH or HTTPS as above).

```
# SSH
git clone git@github.com:username/repo-name.git

# HTTPS
# git clone https://github.com/username/repo-name.git
```

7 Basic Workflow

Navigate to the repo folder:

```
cd repo-name
```

Make changes, then use the following Git commands:

```
git add .                % Stage all changes
git commit -m "Message"  % Commit with message
git push                 % Push to remote repo
```

8 Pulling Changes

To update your local repo with remote changes:

```
git pull
```

9 Creating a Branch

Branches allow you to work on features independently:

```
git checkout -b feature-branch
```

10 Working With Personal Branches

In this course, we will use the following GitHub repository:

`https://github.com/setinski/Leiden-2025`

Each student is required to:

- Clone the repository to their local machine.
- Create a personal branch named after themselves (e.g., `Mickey`).
- Upload their solutions to the appropriate directories within their branch.

To create and switch to a personal branch:

```
git checkout -b your-name
git push --set-upstream origin your-name
```

After adding your solutions and committing:

```
git add .
git commit -m "Add my solutions"
git push origin your-name
```

11 Keeping Your Branch Updated

When you are working on a personal branch, you should regularly pull updates from the main branch:

```
# Make sure you are on your branch
git checkout your-name

# Option 1: Fetch + Merge
git fetch origin
git merge origin/main

# Option 2: Use pull directly
git pull origin main
```

12 Switching Branches

You can always switch back to the main branch using:

```
git checkout main
```