



## 07. Versioning, Git

 **Warning**

**Test 2 on December 8.**

### Lecture

#### Version control, Git



- Track changes in a set of files
- Coordinating work among developers
- Who made what changes and when
- Revert back at any time
- Local and remote repos
- Take snapshots of files by making a *commit*

#### Install

```
sudo apt install git
```

## Basic commands

```
git init      # Initialize local git repo
git add <file> # Add file/files to staging area
git status    # Check status of working tree and staging area
git commit -m "What I've done" # Commit changes in index
git push      # Push to remote repository
git pull      # Pull latest changes from remote repo
git branch <new_branch_name>
git checkout <branch_name>
git merge <branch_name> # Merge the branch into the current branch
git config --global user.name "Istvan Szabo"
git config --global user.email "istvan.szabo@gmail.com"
```

### Tip

**Store personal token:** `git config --global credential.helper store`

### Tip

**Windows and Linux clock issue:** `timedatectl set-local-rtc 1 --adjust-system-clock`

## GitHub



```
git remote
git clone <link> # Copy repo into a new directory

# Add remote to repository:
```

```
git remote add origin <link>  
git push -u origin master
```

### Some alternatives to GitHub

GitLab, BitBucket, Launchpad, Phabricator

## Markdown

- Markup language, easy to read
- Text file → Formatted document
- Widespread usage, e.g., blogs, forums, documentations, readme files, GitHub
- [Markdown Cheatsheet](#)

## Gyakorlat

### 0: GitHub repo létrehozása

1. Inicializáljunk egy lokális git repo-t a `ros-course` package-ben.
2. Regisztráljunk GitHub-ra, majd hozzunk létre egy private repo-t a `ros_course` package számára. Állítsuk be a local repo-ban a remote-ot, majd push-oljuk a package tartalmát.