

Avanceret Programming (Uge 40)

Christian Gram Kalhauge (CKL)

Today

- Project
- Git / Github
- Sorting

Section 1

Project

Project

100 hours project.

- Due December 10th
- Deliveries and presentations (5 min) every 2 weeks:
 - October 8th
 - October 29th
 - November 12th
 - November 26th
- Delivery: Github-link with README.md.
- Exam: Presentation of project with follow-up questions.

Section 2

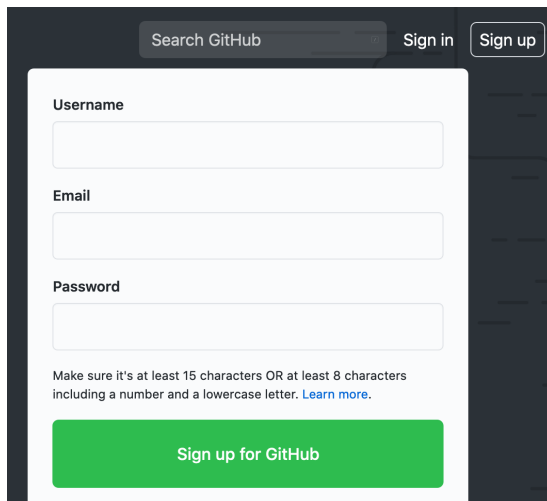
Git / Github

Git

Git is a Version Control System. It is almost impossible to do any kind of development without it.

Github - Create User

Create a user on GitHub



The image shows the GitHub sign-up interface. At the top, there is a dark header with a search bar labeled "Search GitHub", a "Sign in" link, and a "Sign up" button. Below this is a white sign-up form with three input fields: "Username", "Email", and "Password". Below the "Password" field, there is a note: "Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)". At the bottom of the form is a large green button labeled "Sign up for GitHub".

Search GitHub Sign in Sign up

Username

Email

Password

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)

Sign up for GitHub

Github - Create Repository



Repositories

 **New**

Find a repository...

Figure 2: Create a Repository

Github - Create Repository (cont.)

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Repository template

Start your repository with a template repository's contents.

No template ▾

Owner *



kalhauge ▾

Repository name *

Great repository names are short and memorable. Need inspiration? How about **verbose-carnival**?

Description (optional)



Public

Anyone on the internet can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☐ **Add a README file**

This is where you can write a long description for your project. [Learn more](#).

☐ **Add .gitignore**

Choose which files not to track from a list of templates. [Learn more](#).

☐ **Choose a license**

Git - Basics

Commit

A specific point in time of the code

Staging

The place you put things to be committed.

Branch

A pointer to a Commit.

Remote

A remote storage of the version history.

Git Bash

Download and install Git Bash from here:

<https://gitforwindows.org/>

Git' Started

Open Git Bash in the folder you'll like the code in.

```
$ git clone <paste-url>
```

The screenshot shows the GitHub interface for the repository 'kalhaugue / python-example'. At the top, there are tabs for 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Wiki', and 'Security'. The 'Code' tab is selected. Below the tabs, there are buttons for 'Go to file', 'Add file', and a green 'Code' button with a download icon. A dropdown menu is open from the 'Code' button, showing options to 'Clone' the repository. The 'Clone' section has three tabs: 'HTTPS', 'SSH', and 'GitHub CLI'. The 'HTTPS' tab is selected, and the URL 'https://github.com/kalhaugue/python-' is visible. A red arrow points to the copy icon next to the URL. Below the URL, it says 'Use Git or checkout with SVN using the web URL.' There is also an option to 'Open with GitHub Desktop'.

kalhaugue / python-example

Unwatch 1

<> Code ⓘ Issues 🔗 Pull requests ⌚ Actions 📁 Projects 📖 Wiki 🛡 Security

master

Go to file Add file Code

Clone

HTTPS SSH GitHub CLI

https://github.com/kalhaugue/python-

Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

kalhaugue Initial commit ...

.gitignore Initial commit

README.md Initial commit

README.md

Git' Setup

```
$ git config --global user.name "Christian Kalhauge"  
$ git config --global user.email ckl@cphbusiness.dk
```

Git' Going

When you have made changes to your repository:

```
# stage all changes  
$ git add .  
# commit the staged changes with a commit message  
# change <commit-message> with your message  
$ git commit -m '<commit-message>'  
# push the results to the remote  
$ git push
```

Git' A Grip

See the status of the repository

```
$ git status
```

See the history

```
$ git log --oneline --graph --all
```

Git' Back

Undo all your changes since last commit:

```
$ git checkout .
```

Go back a commit

```
$ git reset --hard HEAD^
```

Go back to the last version you uploaded

```
$ git reset --hard origin/master
```


Git' Good

<https://www.youtube.com/watch?v=2sjqTHE0zok>

<https://git-scm.com/book/en/v2/>

<https://davidenunes.com/git-good/>

Section 3

Sorting

Test if a list is sorted – Exercise

`issorted.py`

Bubblesort - Exercise

https://en.wikipedia.org/wiki/Bubble_sort

`bubblesort.py`

Merge - Exercise

`merge.py`

Mergesort - Exercise

https://en.wikipedia.org/wiki/Merge_sort
`mergesort.py`

Quick-Sort

<https://en.wikipedia.org/wiki/Quicksort>

`quicksort.py`