Computer Science Fundamentals

Seminar 2

Seminar objectives

Part A

1. Applications types and their suitability for purpose
2. Software review sites
3. Trello, Miro

Part B

4. Practise basic cmd commands
5. Practise basic git commands
6. Practise basic python

Part A

Applications types and their suitability

Do you use telegram to communicate when you work in group?

Imagine a busy (software or any other product) production team, that has intensive communication load on daily basis to deliver multiple products (majority products being developed in parallel and different speed). Will Telegram suit them?

Applications types and their suitability

How to find needed application(s) among the multitude of available options?

1. Identify general categories of needed software
2. Consider number of users, business requirements, software features
3. Consider budget limitations
4. Consider compatibility with existing OS and/or related software
5. Research what software do companies similar to yours use
6. Read reviews and look for alternatives on sites like:
   a. https://www.g2.com/
   b. https://alternativeto.net/
   c. https://www.capterra.com/ and many others
7. Compile initial set of options and compare software solutions

G2Crowd.com

Explore

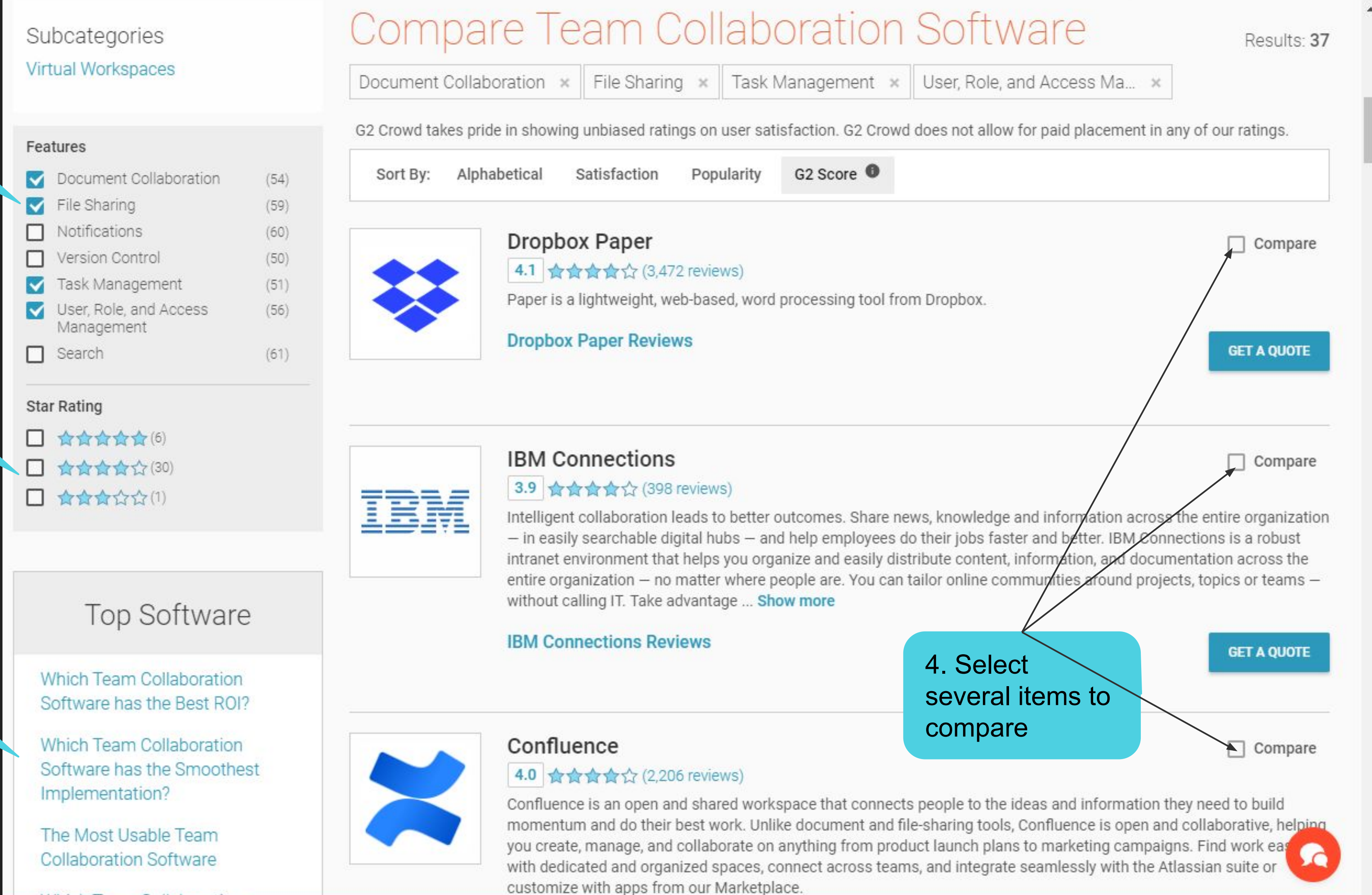
- https://www.g2.com/ > Software > Collaboration and Productivity>Team collaboration > Screen Sharing software

1. Filter the solutions by features

2. Filter the solutions by rating

3. Checkout recommendations

4. Select several items to compare



Subcategories

Virtual Workspaces

Features

- [✓] Document Collaboration (54)
- [✓] File Sharing (59)
- [ ] Notifications (60)
- [ ] Version Control (50)
- [✓] Task Management (51)
- [✓] User, Role, and Access Management (56)
- [ ] Search (61)

Star Rating

- [ ] ★★★★★ (6)
- [ ] ★★★★☆ (30)
- [ ] ★★★☆☆ (1)

Top Software

Which Team Collaboration Software has the Best ROI?

Which Team Collaboration Software has the Smoothest Implementation?

The Most Usable Team Collaboration Software

Compare Team Collaboration Software

Results: **37**

Document Collaboration × | File Sharing × | Task Management × | User, Role, and Access Ma... ×

G2 Crowd takes pride in showing unbiased ratings on user satisfaction. G2 Crowd does not allow for paid placement in any of our ratings.

Sort By: Alphabetical | Satisfaction | Popularity | G2 Score

Dropbox Paper

4.1 ★★★★☆ (3,472 reviews)

Paper is a lightweight, web-based, word processing tool from Dropbox.

Dropbox Paper Reviews

Compare

GET A QUOTE

IBM Connections

3.9 ★★★★☆ (398 reviews)

Intelligent collaboration leads to better outcomes. Share news, knowledge and information across the entire organization — in easily searchable digital hubs — and help employees do their jobs faster and better. IBM Connections is a robust intranet environment that helps you organize and easily distribute content, information, and documentation across the entire organization — no matter where people are. You can tailor online communities around projects, topics or teams — without calling IT. Take advantage ... Show more

IBM Connections Reviews

Compare

GET A QUOTE

Confluence

4.0 ★★★★☆ (2,206 reviews)

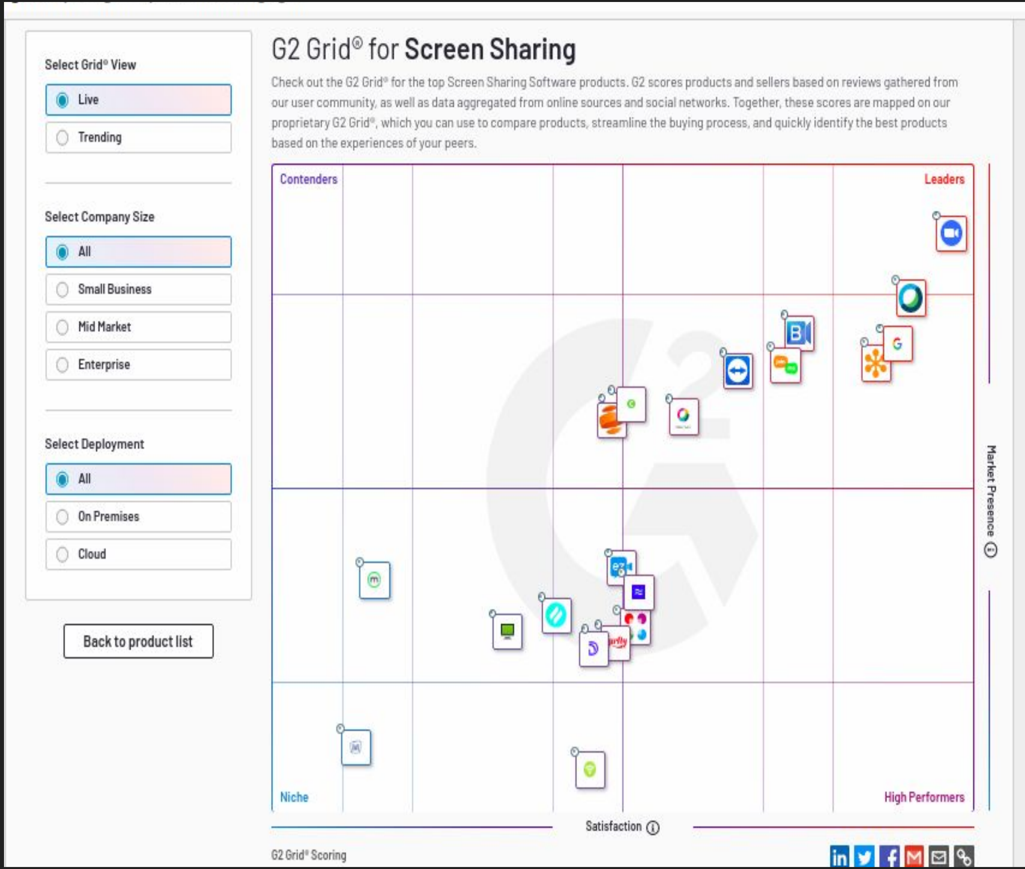
Confluence is an open and shared workspace that connects people to the ideas and information they need to build momentum and do their best work. Unlike document and file-sharing tools, Confluence is open and collaborative, helping you create, manage, and collaborate on anything from product launch plans to marketing campaigns. Find work eas with dedicated and organized spaces, connect across teams, and integrate seamlessly with the Atlassian suite or customize with apps from our Marketplace.

Compare

G2Crowd.com

https://www.g2.com/categories/screen-sharing#grid
Explore G2Crowd Grid:

- Satisfaction and Market presence
  - **Leaders** (higher satisfaction score, higher market presence)
  - **Contenders** (lower satisfaction score, higher market presence)
  - **High performers** (higher satisfaction score, lower market presence)
  - **Niche** (lower satisfaction score, lower market presence)
- Search for particular product and locate it on grid
- Filter solutions by categories (All segments, Small business and etc)
- Click on solution icons on the grid and access to reviews



Select Grid® View
[✓ Live]
Trending

Select Company Size
[✓ All]
Small Business
Mid Market
Enterprise

Select Deployment
[✓ All]
On Premises
Cloud

Back to product list

G2 Grid® for **Screen Sharing**

Check out the G2 Grid® for the top Screen Sharing Software products. G2 scores products and sellers based on reviews gathered from our user community, as well as data aggregated from online sources and social networks. Together, these scores are mapped on our proprietary G2 Grid®, which you can use to compare products, streamline the buying process, and quickly identify the best products based on the experiences of your peers.

G2 Grid® Scoring

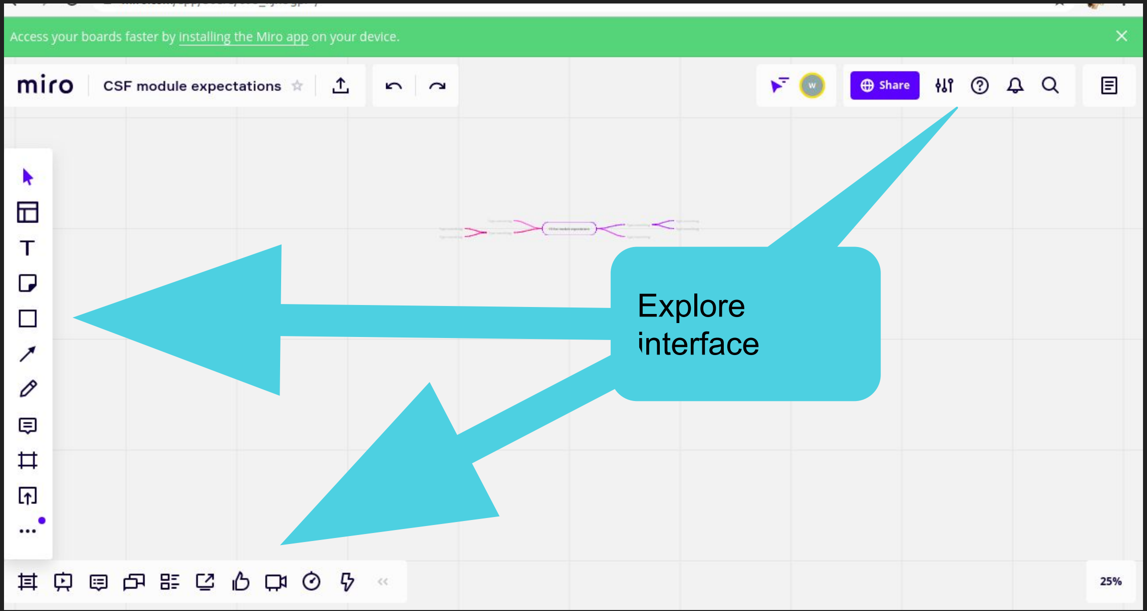
Let’s try Miro

1. Access the following shared Miro board

   Board for groups 1, 2, 3, 4

   Board for groups 5, 6 7

2. Register on Miro
3. Explore interface
4. Let’s collaborate and list module expectations

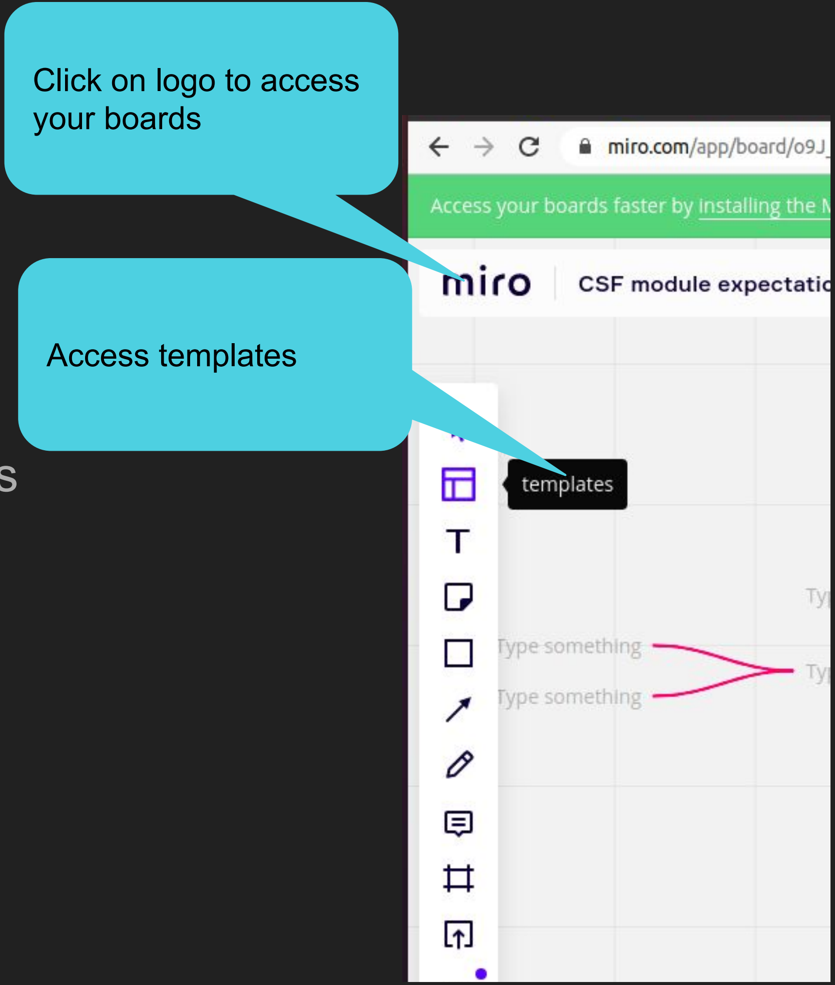

Access your boards faster by installing the Miro app on your device.

Miro

Create your own board

Access templates

Experiment with sharing options



Homework 1: Try Trello

Watch introduction into Trello

Register on Trello using your gmail account (student id based) and plan a trip or any other desired activity in pair. Share with wiut.tutor@gmail.com

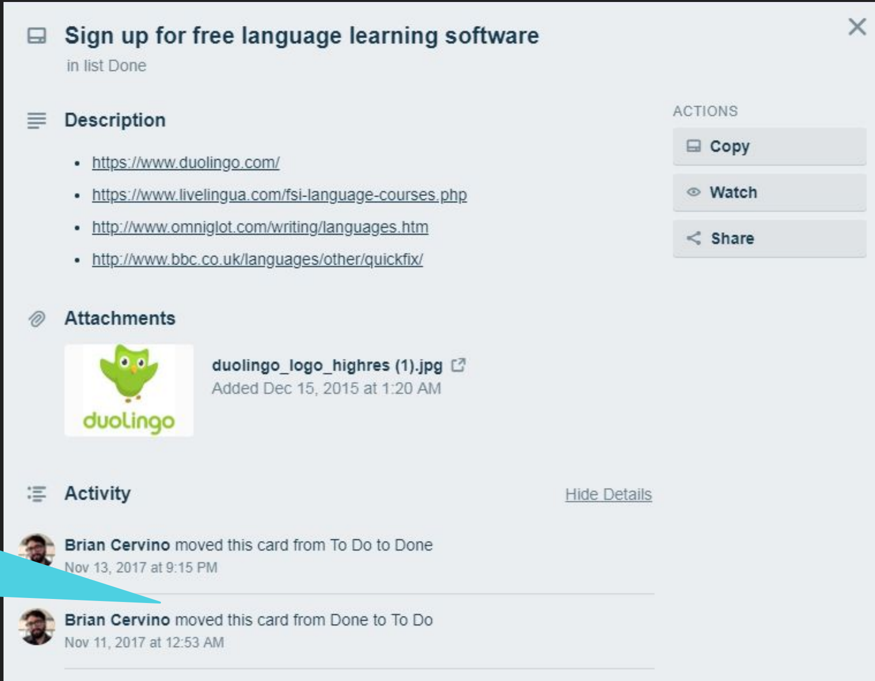
Try:

- Change of the status of the task by moving it to a relevant board (planned ->in progress -> done)
- Assign the task
- Set deadline of the task
- Color code the task

Homework 1: Trello review ready boards

1. Review examples of Trello boards
   a. https://trello.com/b/ijCThESo/goal-learn-a-language
   b. https://trello.com/inspiringboards

Note the status of the card can be monitored.
So when you use Trello for CW planning we will also be able to monitor the progress. Even if all cards will end up in Done list, we will see the history of changed card status.


Sign up for free language learning software
in list Done

Description

- https://www.duolingo.com/
- https://www.livelingua.com/fsi-language-courses.php
- http://www.omniglot.com/writing/languages.htm
- http://www.bbc.co.uk/languages/other/quickfix/

Attachments

duolingo_logo_highres (1).jpg
Added Dec 15, 2015 at 1:20 AM

Activity

Brian Cervino moved this card from To Do to Done
Nov 13, 2017 at 9:15 PM

Brian Cervino moved this card from Done to To Do
Nov 11, 2017 at 12:53 AM

Homework 1: more on Trello

Video on how to use Trello for beginners

Using Trello - Reference

10 Ways to Use Trello

Part B

A bit of history...1980s

Before Windows OS, the most common operating system that ran on IBM PC compatibles was DOS (Disk Operating System)

- not graphical, purely textual
- command line interface
- to run programs or manipulate the operating system you had to type in commands

When Windows was first created it was actually a graphical user interface that was created in order to make using the DOS operating system easier for a novice user

Command prompt a.k.a. a shell, console window, a cmd prompt

Though the newer operating systems do not run on DOS, they do have the command prompt (cmd.exe), which has a similar appearance to DOS.
Other OS families like Macintosh, Unix, Linux have slightly different analogue to Windows command prompt under different name terminal.

So why bother with cmd if we have graphical interface?

- To become expert computer user
    - It enables you to work faster
    - Basic commands are needed while using Git

Command prompt or terminal

WIndows:

- Click on the Start Menu
- Search for cmd.exe

OR

Ubuntu:

Press Ctrl+Alt+T or

Click on Show applications icon and search for Terminal

The command prompt is simply a window that by default displays the current directory, or in windows term a folder, that you are in and has a blinking cursor ready for you to type your commands.
To use the command prompt you would type in the commands and instructions you want and then press enter.

Help command

This command lists all the commands built into the command prompt. If you would like further information about a particular command you can type help command name. For example help cd will give you more detailed information on a command **dir**.

In command prompt:

1. Type help
2. Select any command from the list and type help [command]
3. E.g. help cd

MKDIR command or MD

Creates directory

Syntax

mkdir [Drive:]Path (for Ubuntu)
md [Drive:]Path

Parameters

Drive: Specifies the drive on which you want to create the new directory.
Path: Required. Specifies the name and location of the new directory. The maximum length of any single path is determined by the file system.
/? : Displays help on the command

Try yourself type in command prompt

```
md /?
md 2020-2021\csf
```

Dir command

Lists the files and directories contained in your current directory, if used without an argument, or the directory you specify as an argument.

Type dir and press enter and you will see

- a listing of the current files in the directory you are in
- information about their file sizes
- date and time they were last written to
- how much space the files in the directory are using and the total amount of free disk space available on the current hard drive.

Dir command continued

Note two directories named . and .., which have special meaning in operating systems.

.	stands for the current directory and
..	stands for the previous directory in the path.

Also note for many commands you can use the * symbol which stands for wildcard.

E.g. typing `dir *.txt` will only list those files that end with .txt.

Cd command or CHDIR

Displays name of or allows you to change your current directory to the one specified

Syntax

chdir [Drive:] [Path]
cd [Drive:] [Path] for Ubuntu
cd [/d] [Drive:] [Path] - changes the drive

Parameters

Drive: Specifies the drive

Path: Required. Specifies the name and location of the new directory. The maximum length of any single path is determined by the file system.

/? : Displays help on the command

Try yourself type in command prompt

```
cd /?
cd 2020-2021\csf
cd..
cd /d e:\users
cls
```

Task 1

Navigate to local **Downloads** folder and list only pdf files.

Task 2

1. Delete CSF folder in previously created directory hierarchy (Z:\2020-2021\CSF) using rmdir command and make semester 1 folder with CSF folder inside

   *In Ubuntu use rm -R

2. Add other modules directories

Task 3

Using MKDIR, MOVE, RMDIR, DEL, COPY, ECHO, MORE commands do the following:

1. Inside CSF folder, create Lectures and Seminars folders
2. Organize previously copied files into relevant folders
3. Make nested directory in CSF/Seminars/seminar2/task2/try1
4. Create a file try1.txt
    a. Run a notepad program by typing notepad.exe
    OR
    b. Write a some text to a try1.txt file in a created directory using echo command
5. Print the content on a screen using more command

Hint: Refer to using help or /? to explore how to use new commands

Git

Let’s review from previous tutorial

What is Git?

What is it used for?

Some terminology

Repository

Initialize

Commit

Status

Configure git

- Open git-cmd.exe
- Type the following commands

```
git config --global user.name "Your name"
git config --global user.email "youremail@gmail.com"
```

You first type "git", followed by a command – "config" and pass an option, which is "--global" in the code above.

The option "--global" means that you set your username and email for Git globally on your computer. No matter how many projects with separate local repositories you create, Git will use the same username and email to mark your commits.

Let’s try it

1. Go to directory

   ```
   cd 2020-2021\CSF\
   md seminar2
   cd seminar2
   ```

2. Type `git init` - to initialize, this command creates new git repository
3. Type `git status` - to check the status of the files
4. Type `echo some text>text.txt` - just for demo purpose
5. `git status` - again, this time note untracked file appeared
6. `git add text.txt`
7. `git commit -m “added files first time”`

Task 4

1. Copy some new files to the directory created (any docs, jpeg, source code, folders)
2. Check status
3. Add files to commit (note you can select only needed files out of all copied to the repository) or use `git add --all` to add all new files in the repository
4. Update text.txt file with new text “new updates from student”
5. Commit changes with message “new updates”

Task 5

Open PyCharm (last seminar you were asked to install it as homework)

If not installed yet, use https://repl.it/languages/python3

Create new project “seminar2”

Write a small program that asks for name and prints “Hello {name}”

Important terms

Variables are simply places to store information and to give that information a name. As the term indicates the information stored is "variable", meaning that it can change

Expressions are combinations of values, variables, and operators that the Python interpreter evaluates to compute a resulting value.

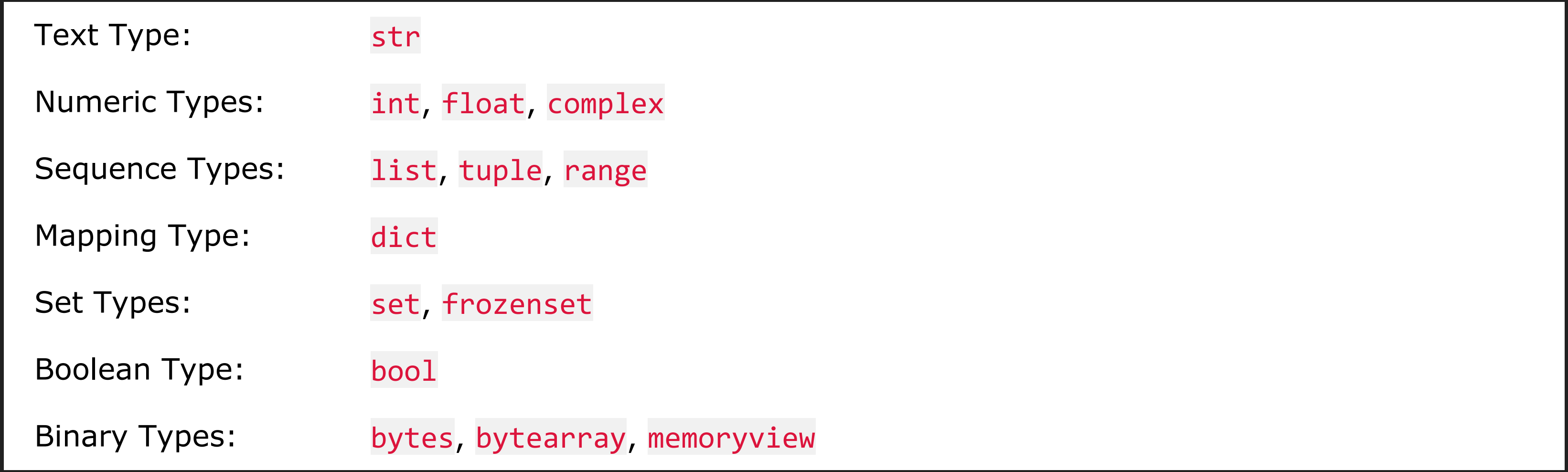
Statements are units of code that have an effect, like creating a variable or displaying a value. The Python interpreter executes statements to produce the effect. (When executing a statement, the interpreter evaluates any expressions included in the statement.)

Functions are named sequences of statements that perform computations. After you define a function, you can call it any number of times to have the Python interpreter execute the statements in the function.

Data types

Variables can store data of different types, and different types do different things.

Python has the following data types built-in by default:



Text Type:	`str`
Numeric Types:	`int`, `float`, `complex`
Sequence Types:	`list`, `tuple`, `range`
Mapping Type:	`dict`
Set Types:	`set`, `frozenset`
Boolean Type:	`bool`
Binary Types:	`bytes`, `bytearray`, `memoryview`

Data types

In Python, the data type is set when you assign a value to a variable:

Data type of any object can be identified by using the type() function:

```
year = 2020

module_name = “CSF”

print(type(year))

print(type(module_name))
```

Expression vs statement

Here are some examples of expressions:

1. `2 + 2`
2. `3 * 7`
3. `1 + 2 + 3 * (8 ** 9) - sqrt(4.0)`
4. `min(2, 22)`
5. `max(3, 94)`
6. `round(81.5)`
7. `"foo"`
8. `"bar"`
9. `"foo" + "bar"`
10. `None`
11. `True`
12. `False`
13. `2`
14. `3`
15. `4.0`

All of the above can be printed or assigned to a variable.

Here are some examples of statements:

1. `if CONDITION:`
2. `elif CONDITION:`
3. `else:`
4. `for VARIABLE in SEQUENCE:`
5. `while CONDITION:`
6. `try:`
7. `except EXCEPTION as e:`
8. `class MYCLASS:`
9. `def MYFUNCTION():`
10. `return SOMETHING`
11. `raise SOMETHING`
12. `with SOMETHING:`

None of the above constructs can be assigned to a variable. They are syntactic elements that serve a purpose, but do not themselves have any intrinsic “value”. In other words, these constructs don’t “evaluate” to anything.

Function

In Python a function is defined using the def keyword:

```
def my_function():
  print("Hello!")
```

The function can be called multiple times by using function name followed by parenthesis

```
my_function()
```

Python task 1

https://intranet.wiut.uz/LearningMaterial/Pages/Details?ID=887&moduleID=0&way=lm

Homework 2: Python task 2

https://intranet.wiut.uz/LearningMaterial/Pages/Details?ID=888&moduleID=0&way=lm

Homework 2:

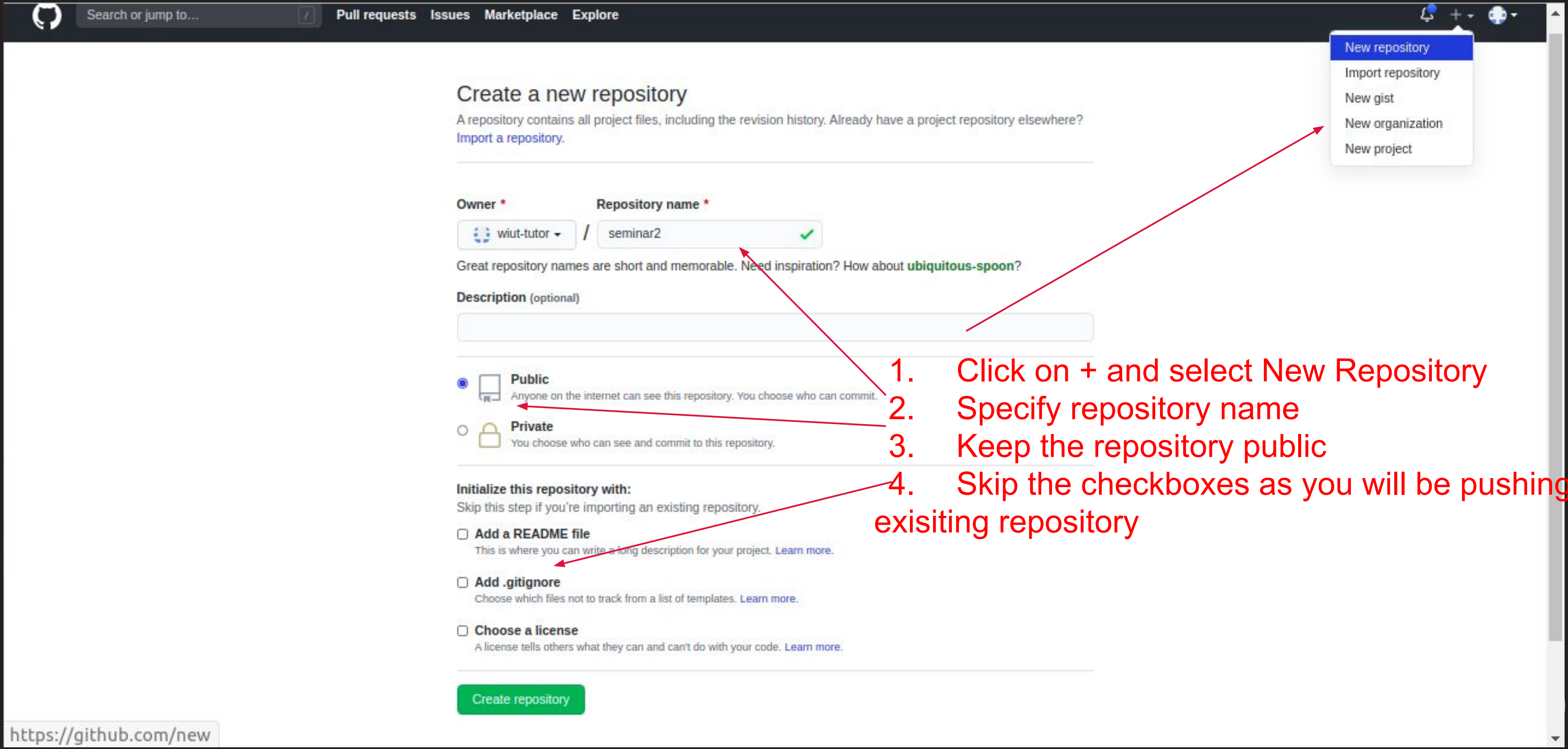
Register on Github with your google account (containing ID, not your name)

Create a new repository and push python tasks 1 and 2 there (see instructions on next slide)

Share your repository link in discussion

https://intranet.wiut.uz/LearningMaterial/Discussion/Details/649?moduleId=559

Github instructions 1



Search or jump to… Pull requests Issues Marketplace Explore

New repository
Import repository
New gist
New organization
New project

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * wiut-tutor / Repository name * seminar2 ✓

Great repository names are short and memorable. Need inspiration? How about ubiquitous-spoon?

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.

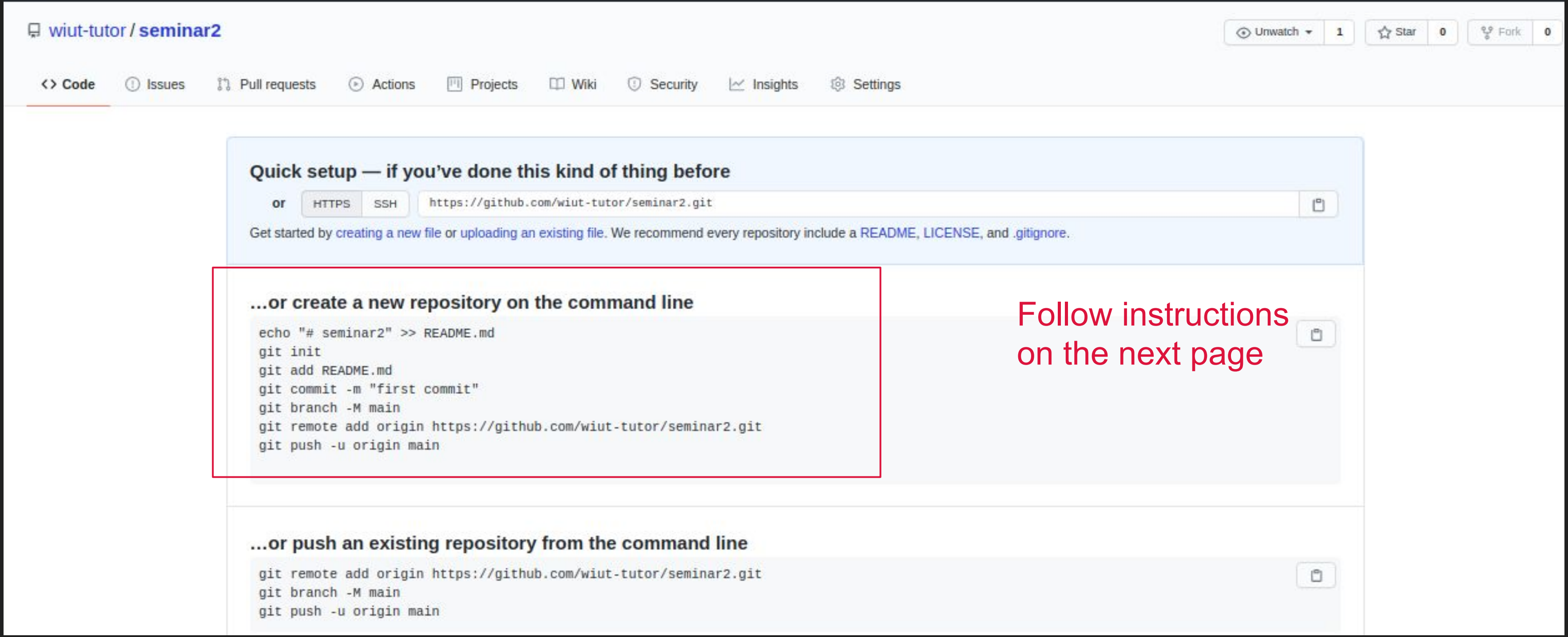
☐ Choose a license
A license tells others what they can and can't do with your code. Learn more.

Create repository

https://github.com/new

1. Click on + and select New Repository
2. Specify repository name
3. Keep the repository public
4. Skip the checkboxes as you will be pushing exisiting repository

Github instructions 2



Quick setup — if you've done this kind of thing before

or HTTPS SSH `https://github.com/wiut-tutor/seminar2.git`

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

```
echo "# seminar2" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/wiut-tutor/seminar2.git
git push -u origin main
```

Follow instructions on the next page

...or push an existing repository from the command line

```
git remote add origin https://github.com/wiut-tutor/seminar2.git
git branch -M main
git push -u origin main
```

Recommended actions

Downey, A. (2015). *Think Python: How to think like a computer scientist.* Green Tea Press. Ch1, 2, 3, available on intranet

Top Software Review Sites + How to Use Them to Find the Perfect App for Your Small Business

For CW up to 5 extra marks **can be** assigned in border-line cases (i.e. 29, 39, 49 and etc) **if** there is evidence of participation in in-class activities and doing h/w

- Read and practice Git
  - http://rogerdudler.github.io/git-guide/
  - https://git-scm.com/about
  - https://rubygarage.org/blog/most-basic-git-commands-with-examples
  - https://ru.atlassian.com/git/tutorials