Non-Research Tips for Information Science Researchers (Summer 2024)

Apr 10, 2024
Week 1: Introduction and Review of Fundamental Concepts

Yusuke Matsui
(UTokyo)

Koya Narumi
(Keio Univ.)

Yuki Koyama
(AIST)

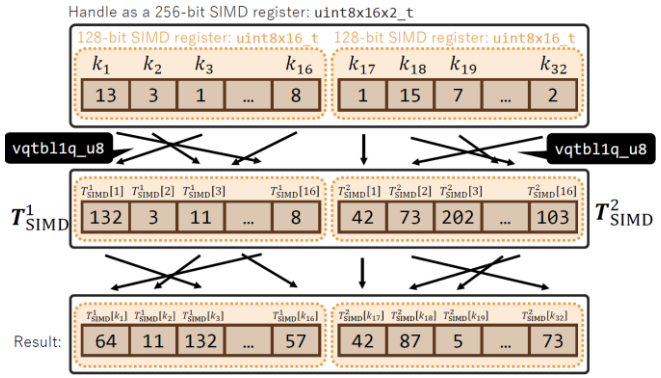
Jun Kato
(AIST)

Yusuke Matsui

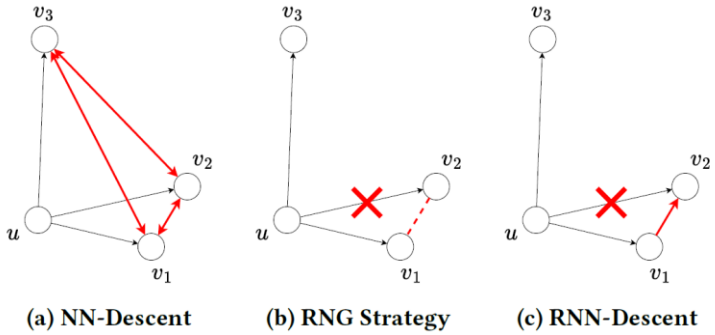
Lecturer (Assistant Professor), the University of Tokyo, Japan

http://yusukematsui.me @utokyo_bunny @matsui528

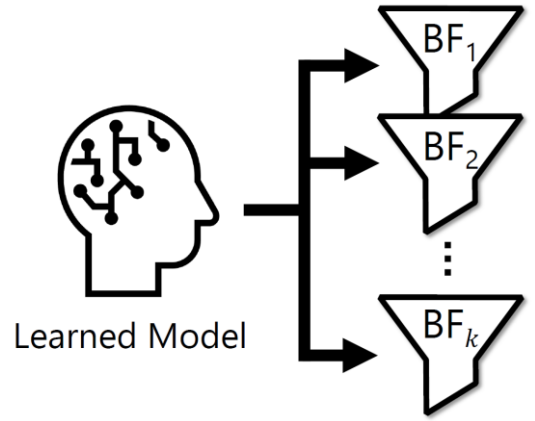
- ✓ Computer vision, Data structure, Machine learning
- ✓ Vector database
- ✓ ML-enhanced DS



ARM impl. of Faiss
[Matsui+, ICASSP 22]



Nearest neighbor search
[Ono & Matsui, ACMMM 23]

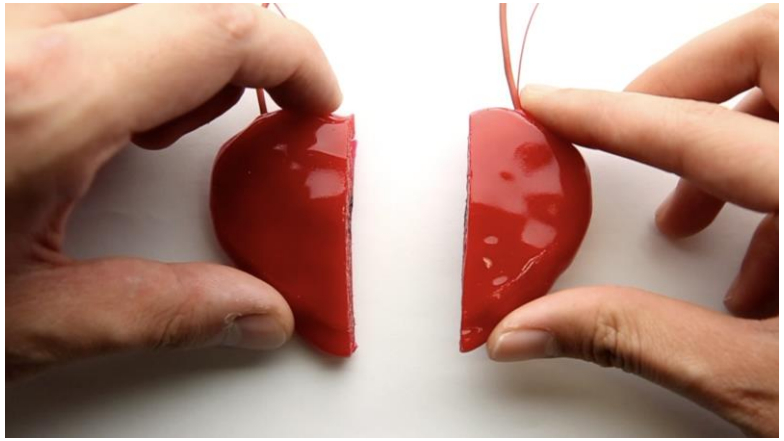


ML-enhanced Bloom Filter
[Sato & Matsui, NeurIPS 23]

Koya Narumi

Associate Professor, Keio University, Japan
https://narumi.me/ @koya_narumi

- ✓ Digital Fabrication
- ✓ Shape-changing Interfaces
- ✓ Human-Computer Interaction



Self-healing UI
[Narumi+, UIST 18]

Inkjet 4D Print
[Narumi & Koyama+, SIGGRAPH 23]

TYPE-X by ISSEY MIYAKE
[Paris Fashion Week 24]

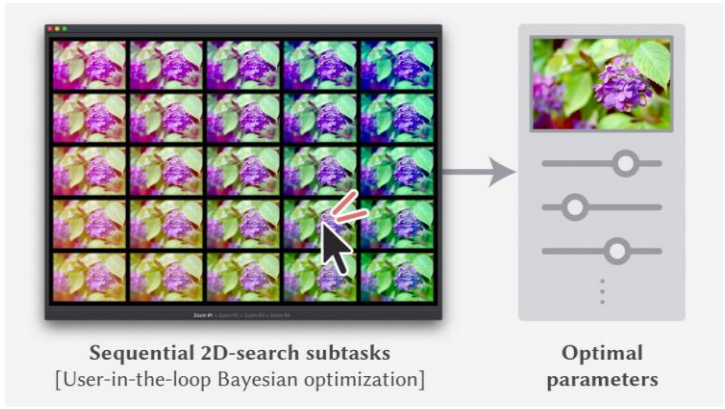
Yuki Koyama

Senior Researcher, AIST, Japan / Technical Advisor, Graphinica, Inc., Japan

https://koyama.xyz @bravery_ @yuki-koyama

- ✓ Computer graphics (and Anime production)
- ✓ Human-computer interaction
- ✓ Human-in-the-loop design optimization

Design suggestion
[Koyama+, UIST 2022]

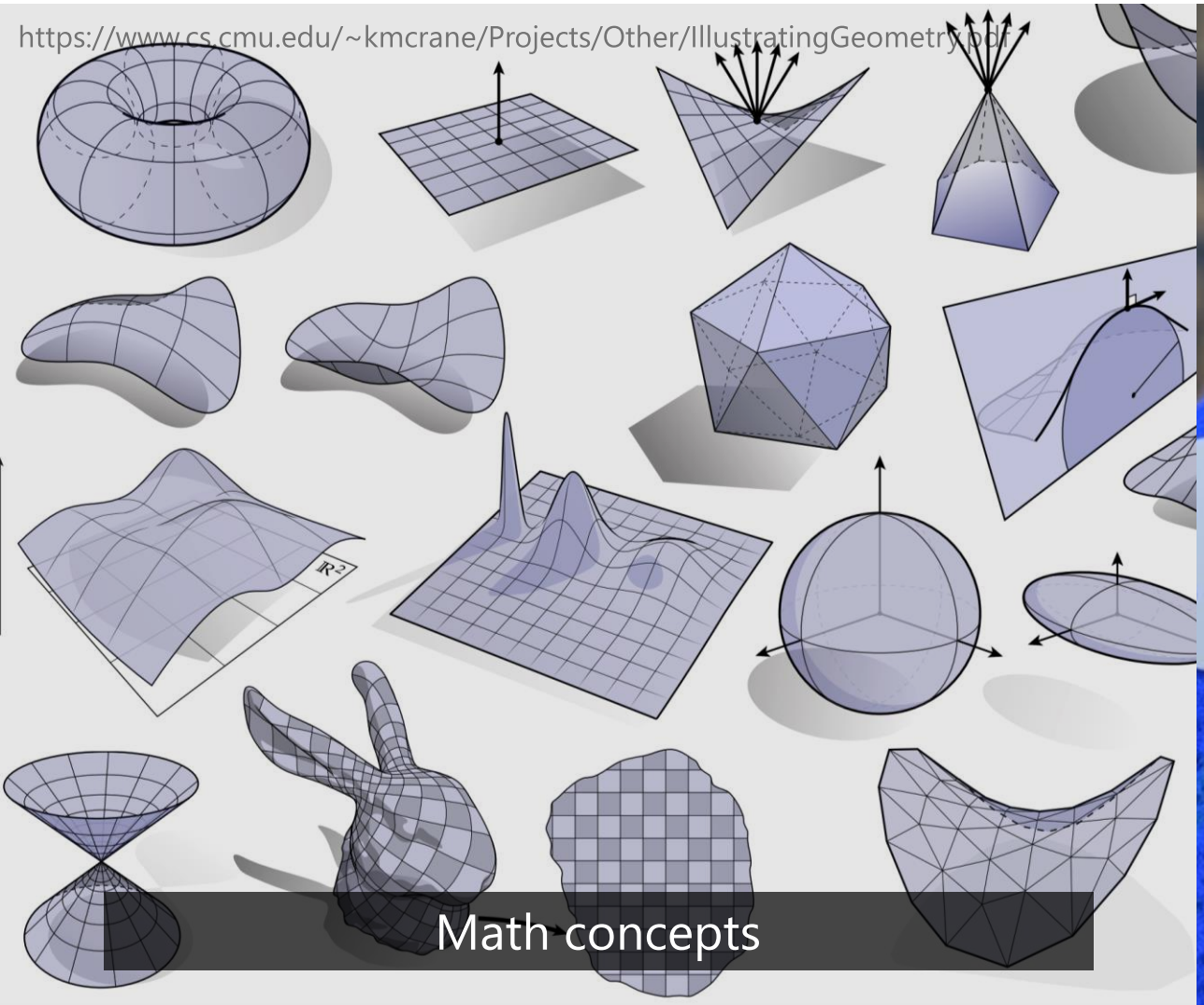


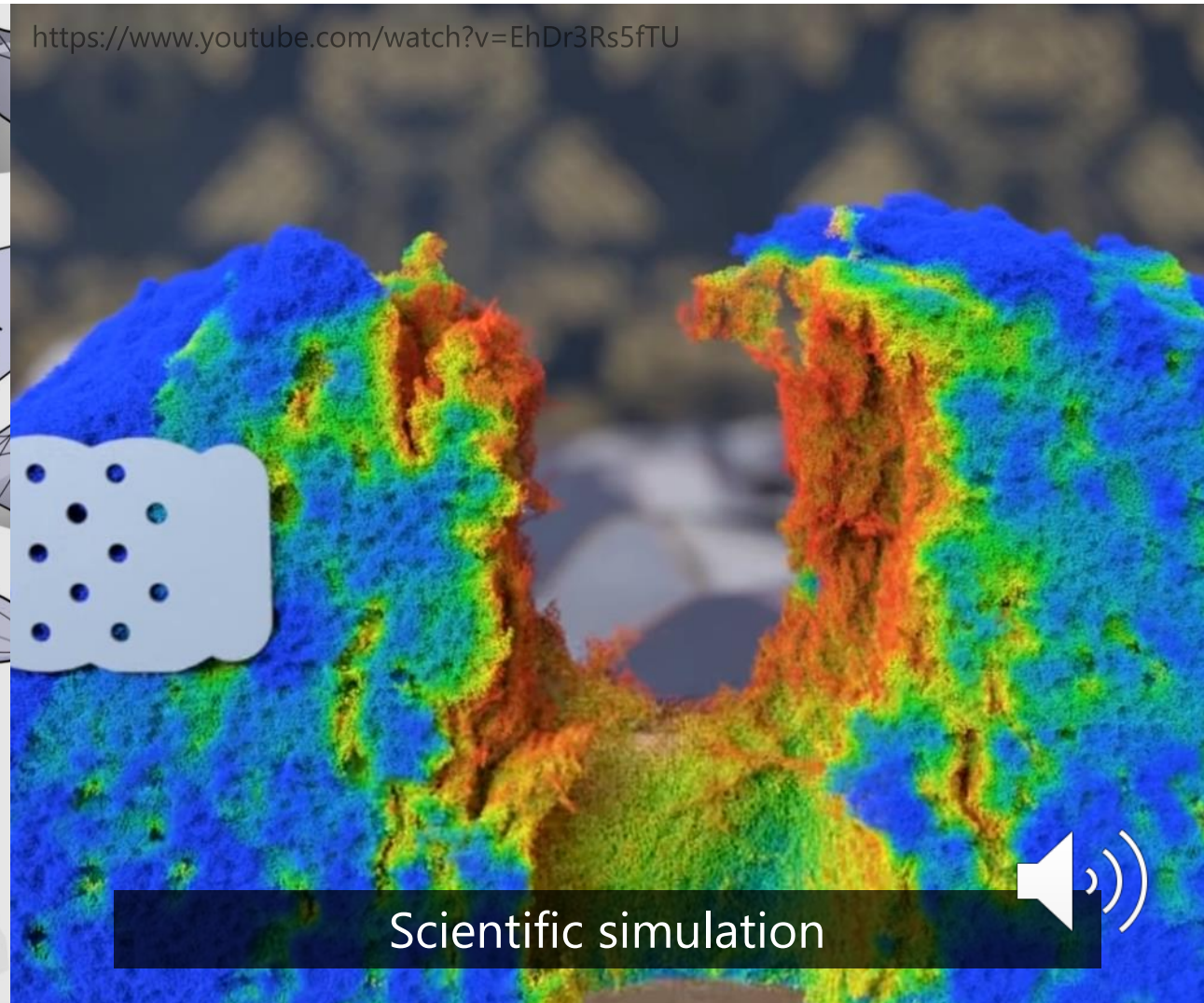
Interactive design optimization
[Koyama+, SIGGRAPH 2020]

Advanced image editing
[Koyama+, CGF 2018]

The topic that I will cover in this course: 3DCG illustrations

Good illustrations help people better understand your research!!
3DCG is sometimes an effective technique for this purpose.

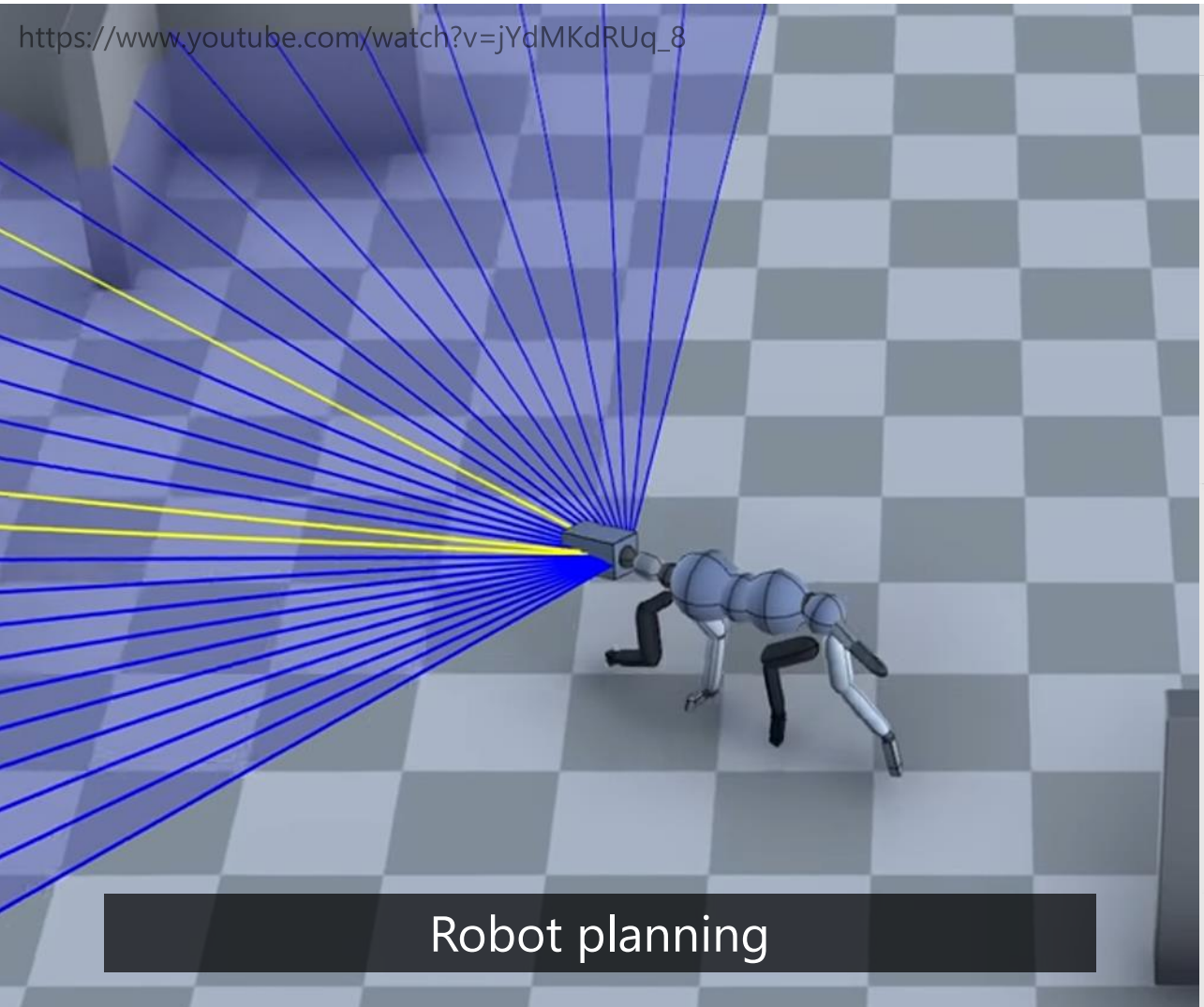




The topic that I will cover in this course: 3DCG illustrations

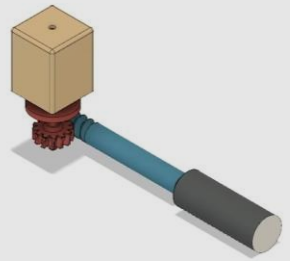
Good illustrations help people better understand your research!!
3DCG is sometimes an effective technique for this purpose.

https://www.youtube.com/watch?v=jYdMKdRUq_8

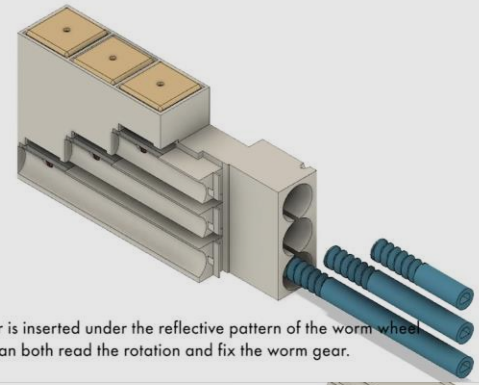


Robot planning

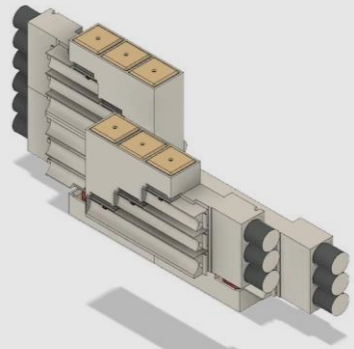
https://www.youtube.com/watch?v=Cd09lh2cXnc



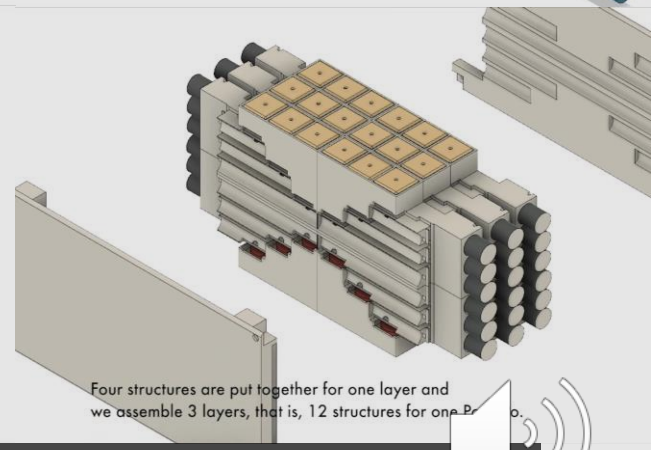
We used worm gear mechanisms to achieve its handheld size.



The spacer is inserted under the reflective pattern of the worm wheel so that it can both read the rotation and fix the worm gear.



Four structures are put together for one layer and we assemble 3 layers, that is, 12 structures for one PoCoPo.



Four structures are put together for one layer and we assemble 3 layers, that is, 12 structures for one P... o.

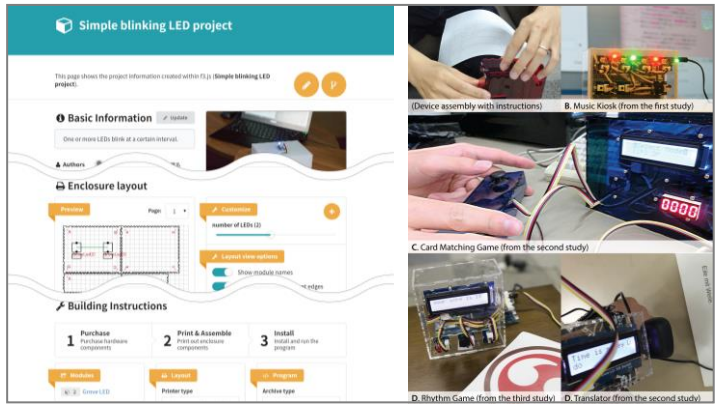
Mechanical assembly

Jun Kato flying to Paris

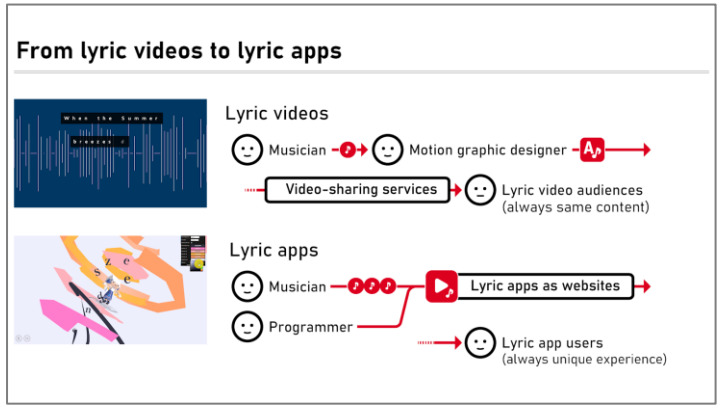
Senior Researcher, AIST, Japan / Technical Advisor, Arch Inc, Japan

https://junkato.jp @junkato @arcatdmz

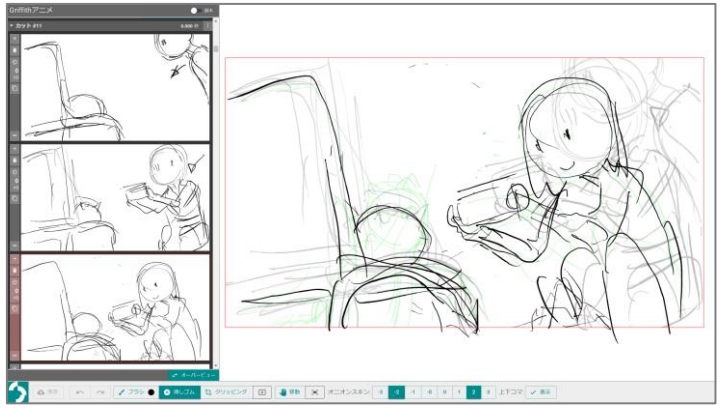
- ✓ Human-Computer Interaction
- ✓ Creativity Support (“Toolsmith researcher” for programmers, musicians, anime creators, etc.)



“f3.js” for parametric device design [Kato+, ACM DIS '17]



“Lyric app” dev framework [Kato+, ACM CHI '23]



Storyboarding tool “Griffith” [Kato+, ACM CHI '24 (to appear)]

People are Programmers 自己紹介 English

https://blog.junkato.jp/ja

People are Programmers

このブログについて

人とコンピュータの関係を考え、よりよくしていく学問 "HCI" 研究者のブログです。**創作支援**のための**ユーザインタフェース**と**統合環境**設計を専門としています。産総研という研究所とアーチというアニメ制作会社に所属しており、研究だけでなく、一般公開サービスの開発や運営も行っています。日々のこと、趣味のこと、いろいろ書きます。

Non-research tips for students
but a bit outdated + in Japanese

加藤 淳

著者について、詳しくは自己紹介へどうぞ。

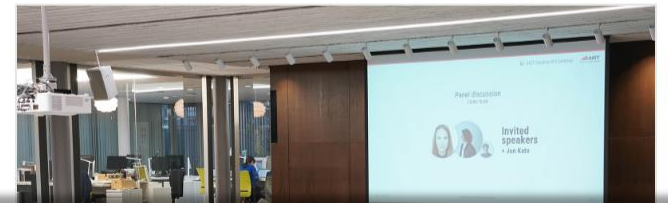
学生向けおすすめ記事

- コンピュータ科学の博士課程にきて初めて分かったこと4つ
- 情報科学系 海外研究インターンのすすめ
- 情報理工学系の産業界とアカデミアは今後どうしたらいいの？
- 情報処理が科学を更新する（IPSJ-ONEに登壇しました）
- 日本からでも応募できる情報科学系Fellowship
- ACM Student Research Competition参加のすすめ
- ACM Doctoral Symposium (Consortium) 参加のすすめ
- 国際会議Student Volunteerのすすめ

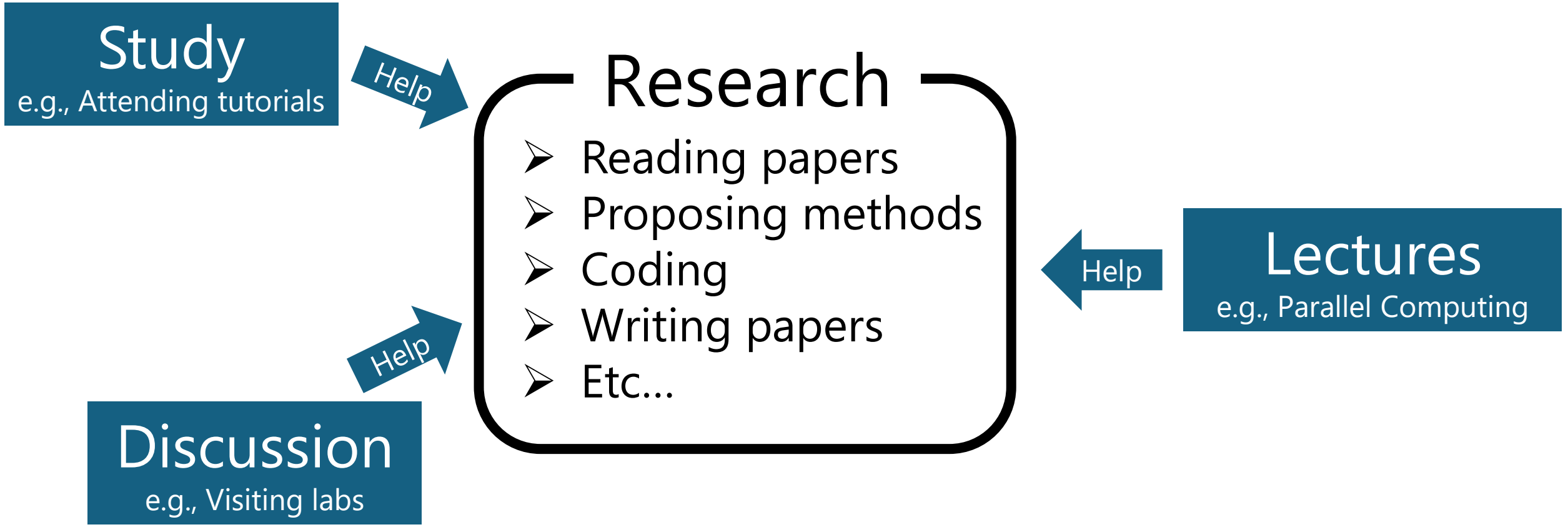
最新の投稿 5 件

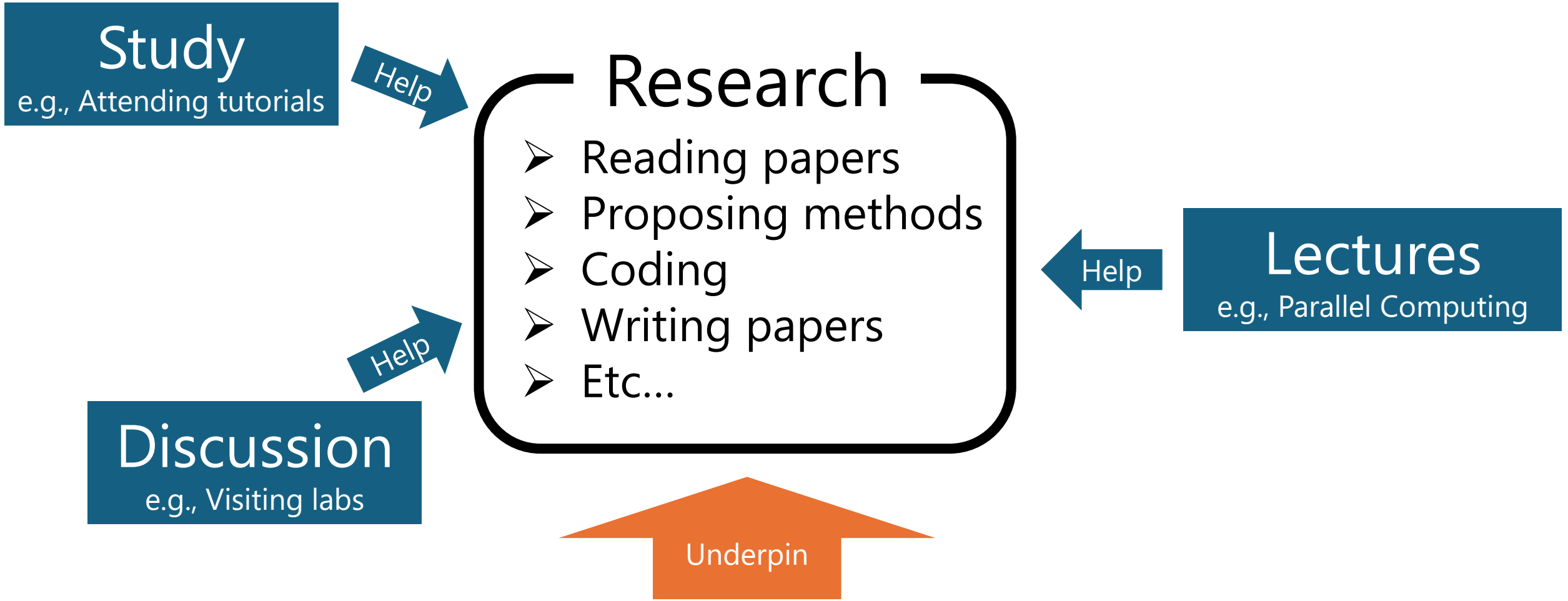
HCI research in the wild, why not?

気付けば元旦になっていましたが、主に国際連携・メタ研究の観点で、2023年を振り返りました。 HCI Advent Calendar 2023 にすでにあがっている記事を読みながら考えたことも併せて書いています。 HCI 研



I hope to have my talks in June that are 1) more practical (web, cloud, CI/CD) and 2) more strategic (how to get involved in the international research community). See you then!


Study
e.g., Attending tutorials
Help
Research
Reading papers
Proposing methods
Coding
Writing papers
Etc…
Help
Lectures
e.g., Parallel Computing
Help
Discussion
e.g., Visiting labs



Important but not taught knowledge

e.g., writing pseudo code, creating a demo video, and managing a research community

- Our target! "**Non-research tips**"
- By mastering these skills, you can focus more on your research itself

Information

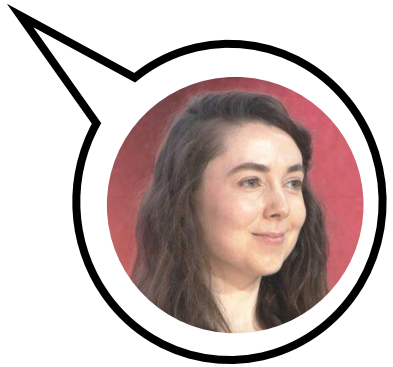
- Dates: Wednesday, 2nd period (10:25 - 12:10)
- Location: Faculty of Engineering Bldg. 2 #244 (This room)
- Style: The lecture will be held in-person. No streaming is available.
- (Japanese materials may also be distributed.)

Assessment

- Plan: Mid-term and/or final reports
- Additionally, those interested (or all participants) may conduct a final presentation (Details are not decided yet)

Schedule

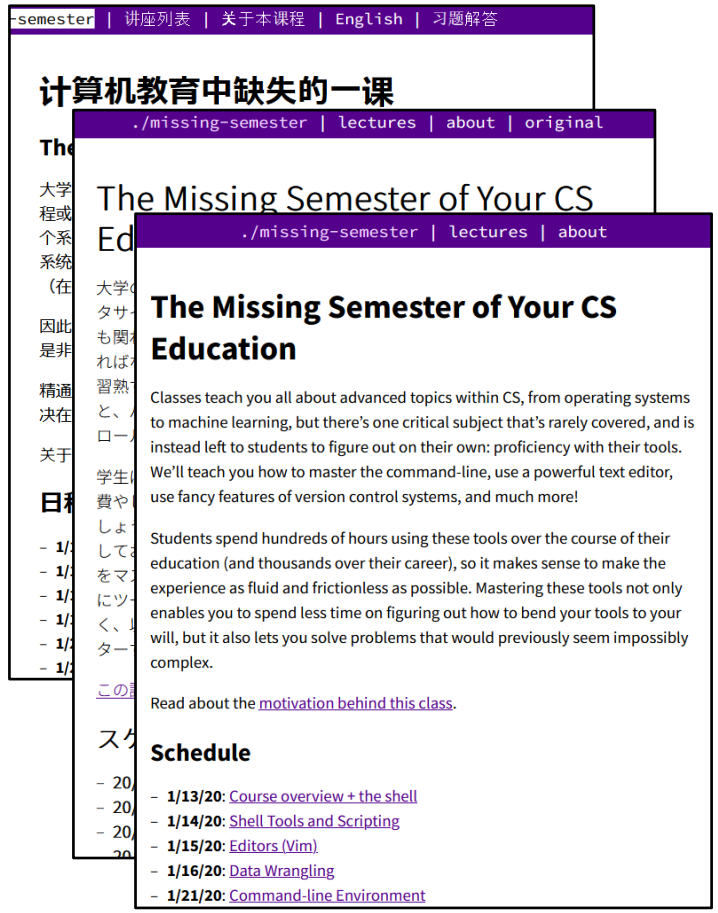
Date (2024)	Contents	Presented by
Week 1, Apr 10	Introduction. Review of fundamental concepts	Yusuke, Koya, Yuki, Jun
Week 2, Apr 17	Equations and pseudo-codes	Yusuke Matsui
Week 3, Apr 24	Presentation	Koya Narumi
Week 4, May 1	Tables and plots	Yusuke Matsui
Week 5, May 8	Figures	Koya Narumi
Week 6, May 22	Videos	Koya Narumi
Week 7, May 29	Invited Talk 1	Dr. Yoshiaki Bando (AIST)
Week 8, June 5	Invited Talk 2	Prof. Katie Seaborn (Tokyo Tech)
Week 9, June 12	GitHub in depth	Yusuke Matsui
Week 10, June 19	Automation of research and research dissemination (Web, Cloud, CI/CD)	Jun Kato
Week 11, June 26	Research community	Jun Kato
Week 12, July 3	3DCG illustrations	Yuki Koyama
Week 13, July 10	Final presentations	-



Prerequisites

The Missing Semester of Your CS Education

- Short lecture series at MIT
  - ✓ https://missing.csail.mit.edu/
- Similar concept to us
  - ✓ Fundamental technical tools
  - ✓ Shell, command-line, git, …
- Translated into several languages
  - ✓ Matsui took charge of the translation into Japanese
- Be sure to read it in advance
- Today's lecture
  - ✓ (1) Review of important concepts of MS
  - ✓ (2) Some more knowledge for UT students



Website

- https://non-research-tips.github.io/2024.html
- We'll upload all materials
- Cancellations, administrative notices, etc., will be announced here

Agenda

- Git / GitHub (basic usage)
- Make
- Docker / Singularity
- Markdown (and structured text description)
- Notebook environment
- LaTeX / Overleaf
- Mental model for computation
- SSH + server coding
- UTokyo services

- You should be familiar with them
- If you don't know any of them, study on your own. There are several online resources available
- I will not cover them in detail in class.

Agenda

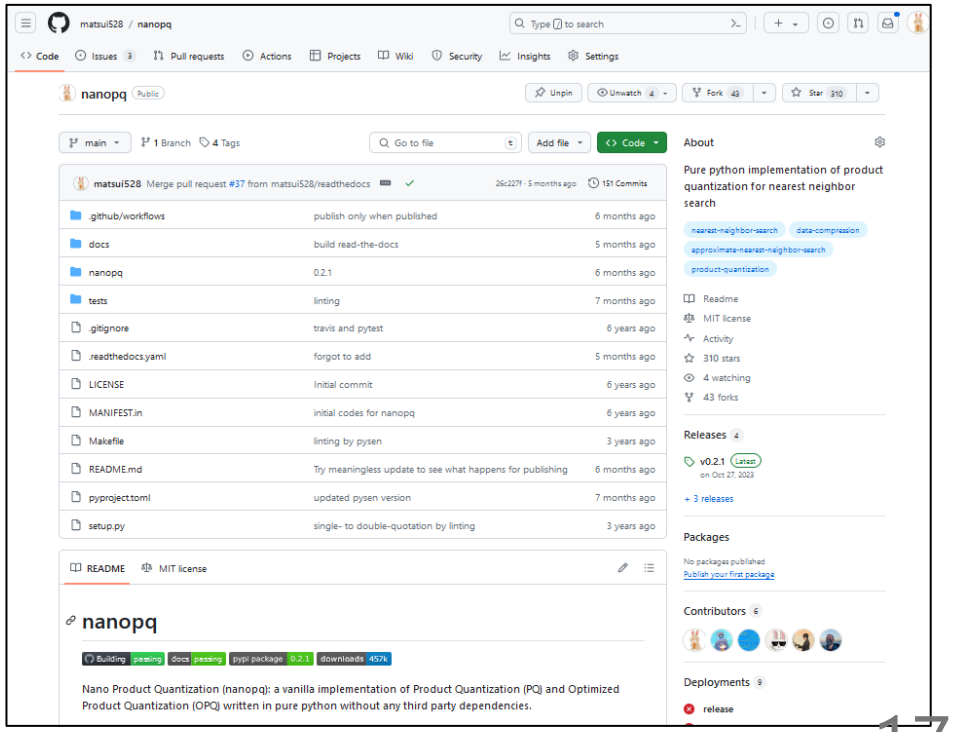
- **Git / GitHub (basic usage)**
- Make
- Docker / Singularity
- Markdown (and structured text description)
- Notebook environment
- LaTeX / Overleaf
- Mental model for computation
- SSH + server coding
- UTokyo services

- You should be familiar with them
- If you don't know any of them, study on your own. There are several online resources available
- I will not cover them in detail in class.

Git / GitHub

- Git
  - ✓ Command for version control: `$ git add xxx.py`
- GitHub
  - ✓ Narrow sense: Webservice to host repositories
  - ✓ Advanced usage: Week 9
  - ✓ Similar services: Bitbucket, GitLab, …

- What you need to know
  - ✓ Backup your code
    - ☐ git add & commit & push
    - ☐ You **MUST** backup your codes!
  - ✓ Collaborations
    - ☐ fork & pull requests & review



Git / GitHub

- Resources (English)
  - ✓ Pro Git: https://git-scm.com/book/en/v2
  - ✓ Learn Git @GitKraken: https://www.gitkraken.com/learn/git

- Resources (Japanese)
  - ✓ GitHub Lecture by Prof. Hiroshi Watanabe https://github.com/kaityo256/github
  - ✓ 渡辺 宙志, “ゼロから学ぶGit／GitHub　現代的なソフトウェア開発のために”, 講談社, 2024 https://www.kspub.co.jp/book/detail/5352199.html

- The easiest way: Ask your friends!

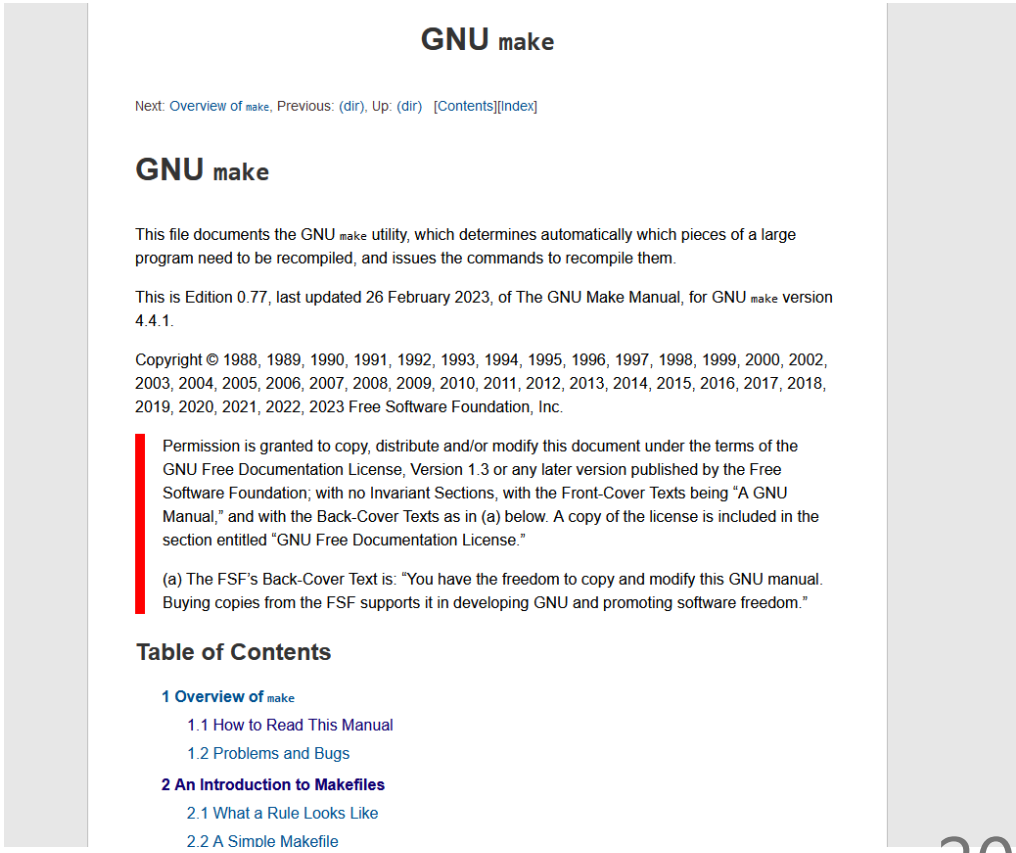
Agenda

- Git / GitHub (basic usage)
- **Make**
- Docker / Singularity
- Markdown (and structured text description)
- Notebook environment
- LaTeX / Overleaf
- Mental model for computation
- SSH + server coding
- UTokyo services

- You should be familiar with them
- If you don't know any of them, study on your own. There are several online resources available
- I will not cover them in detail in class.

Make

- General purpose workflow description (not just compiling c/c++)
- Manual: https://www.gnu.org/software/make/manual/make.html
- Makefile highlights:
  - ✓ What users can (should) do
  - ✓ The order of operations
- The only tool everyone can understand
  - ✓ 🤔 cmake? Bazel? Meson?
- Can be used for **general** purposes



GNU make

Next: Overview of make, Previous: (dir), Up: (dir) [Contents][Index]

GNU make

This file documents the GNU make utility, which determines automatically which pieces of a large program need to be recompiled, and issues the commands to recompile them.

This is Edition 0.77, last updated 26 February 2023, of The GNU Make Manual, for GNU make version 4.4.1.

Copyright © 1988, 1989, 1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023 Free Software Foundation, Inc.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, with the Front-Cover Texts being "A GNU Manual," and with the Back-Cover Texts as in (a) below. A copy of the license is included in the section entitled "GNU Free Documentation License."

(a) The FSF's Back-Cover Text is: "You have the freedom to copy and modify this GNU manual. Buying copies from the FSF supports it in developing GNU and promoting software freedom."

Table of Contents

1 Overview of make
1.1 How to Read This Manual
1.2 Problems and Bugs
2 An Introduction to Makefiles
2.1 What a Rule Looks Like
2.2 A Simple Makefile

Agenda

- Git / GitHub (basic usage)
- Make
- **Docker / Singularity**
- Markdown (and structured text description)
- Notebook environment
- LaTeX / Overleaf
- Mental model for computation
- SSH + server coding
- UTokyo services

- You should be familiar with them
- If you don't know any of them, study on your own. There are several online resources available
- I will not cover them in detail in class.

Docker / Singularity

https://www.docker.com/ja-jp/company/newsroom/media-resources/

- A de facto standard tool for virtualization (container)

- In modern software engineering
  - ✓ One would like to perfectly use the same runtime environment even 1 month later, by different OS, by different hardware
  - ✓ Virtualization is the only option; Docker won the *VM war*

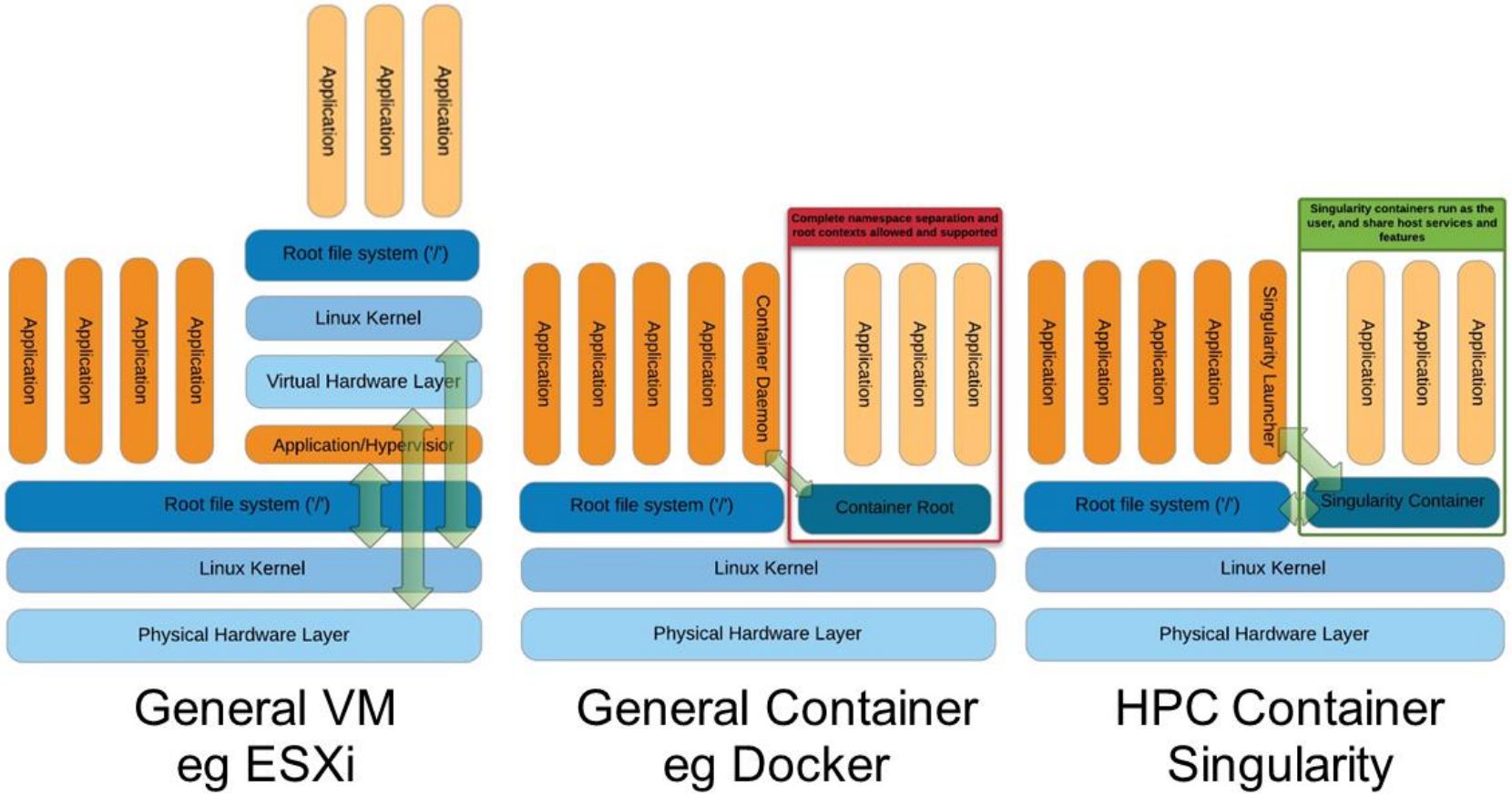
- In research
  - ✓ Reproducibility is the key
  - ✓ Docker provides the easiest way to reproduce your results

- Docker desktop is freely available for academic purpose
  - ✓ Not free for a big company. Be careful!

Docker / Singularity

- tldr; Docker for HPC
- The default option for ABCI is Singularity

https://sylabs.io/docs/



Source: Greg Kurtzer keynote at HPC Advisory Council 2017 @ Stanford

Agenda

- Git / GitHub (basic usage)
- Make
- Docker / Singularity
- **Markdown (and structured text description)**
- Notebook environment
- LaTeX / Overleaf
- Mental model for computation
- SSH + server coding
- UTokyo services

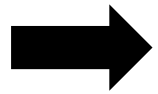
- You should be familiar with them
- If you don't know any of them, study on your own. There are several online resources available
- I will not cover them in detail in class.

Markdown (and structured text description)

```
# This is H1 title

## This is H2 title
usual text

- list item 1
- list item 2
- list item 3

this is [link](https://www.google.com)
```

README.md

Render



This is H1 title

This is H2 title

usual text

- list item 1
- list item 2
- list item 3

this is link

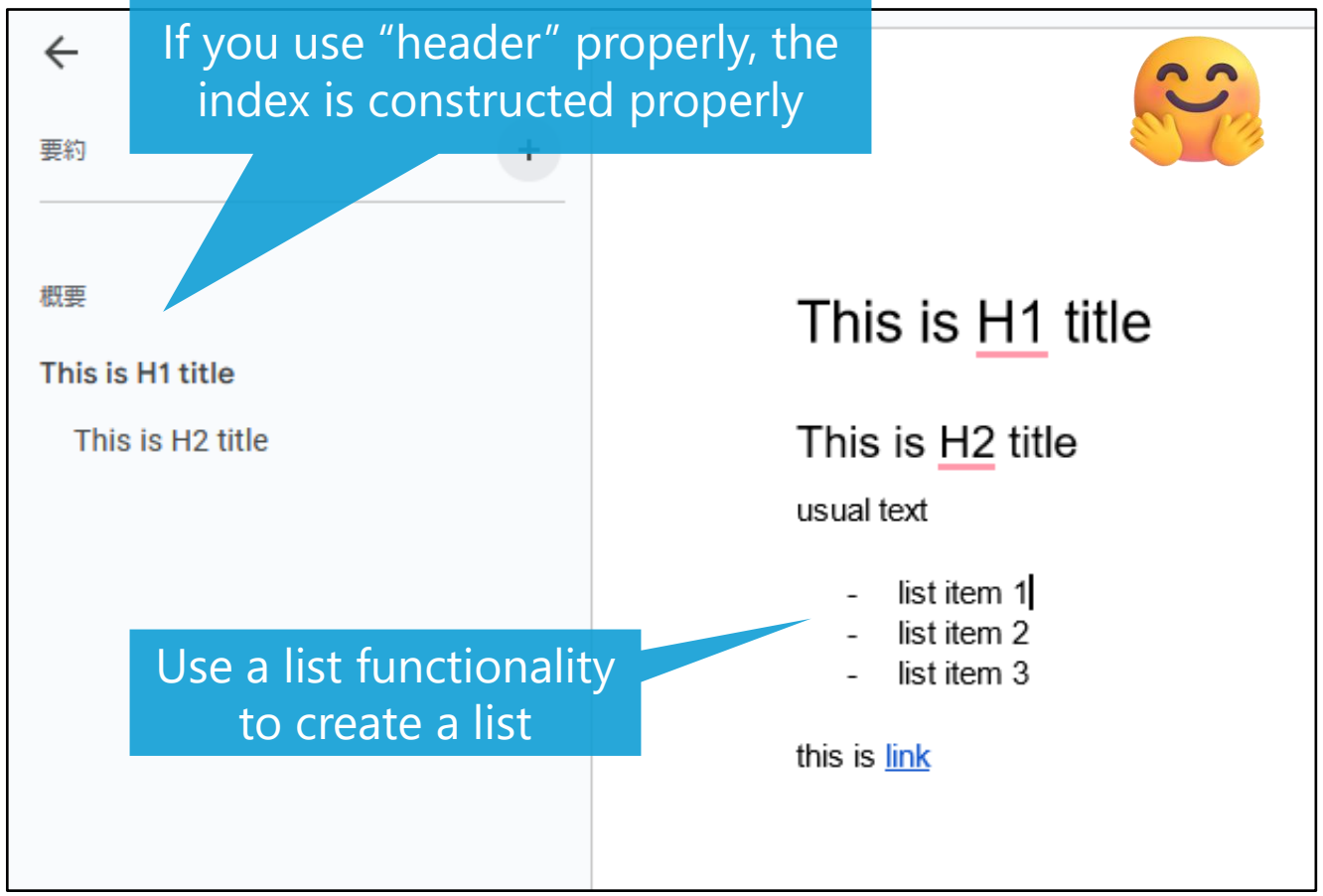
- Programmable (version-controrable), structured text
- Computers like Markdown; easily reusable for later purposes
  - ✓ e.g., convert to tex, pdf, html, etc…
- Structured texts are important to communicate with others

Markdown (and structured text description)

- Advanced usage for GitHub markdown
  - ✓ Copy & paste images
  - ✓ Diagrams
  - ✓ Equations
  - ✓ Alerts

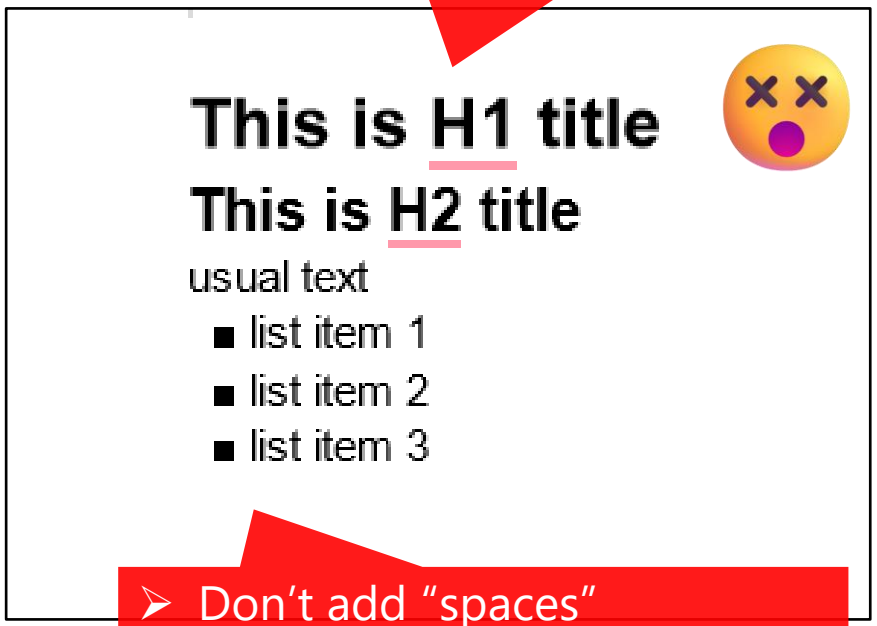
Markdown (and structured text description)

- Recommend: write google-doc and MS word as if it's markdown



Don't change the "visual" thing.
The text should be structured.

Don't make a header by changing the font size!



- Don't add "spaces"
- Don't create your own bullets

Why bad? It's hard to maintain by others!

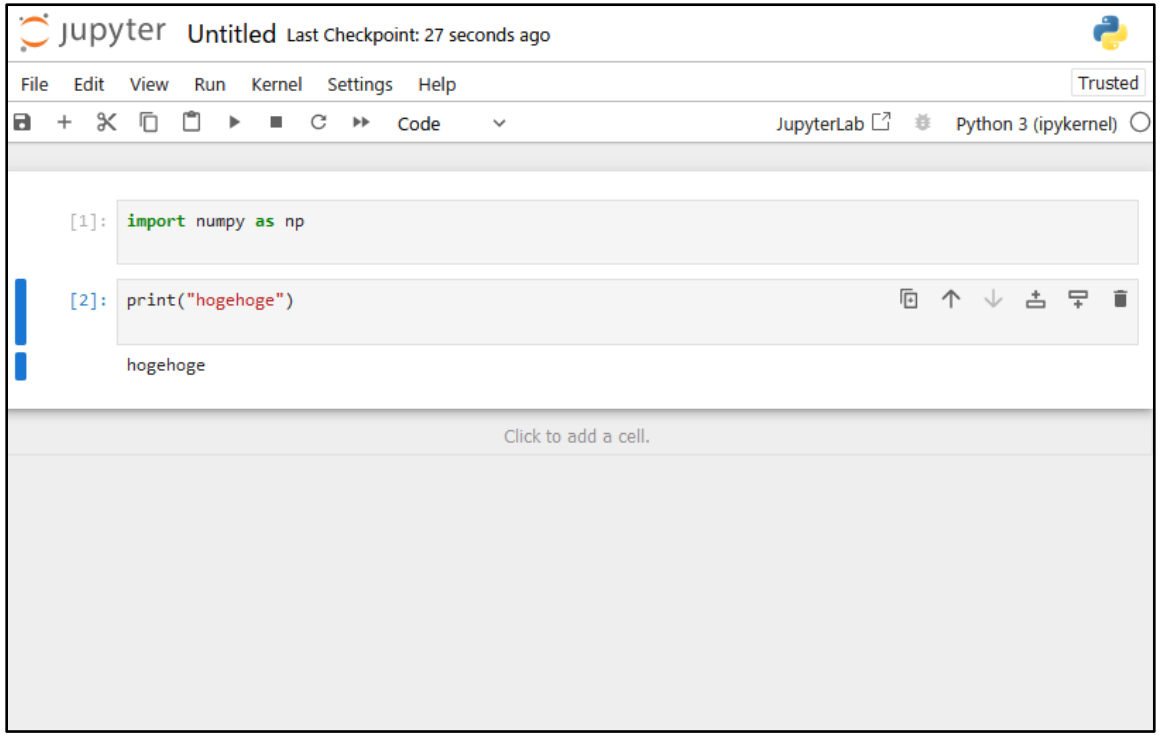
Agenda

- Git / GitHub (basic usage)
- Make
- Docker / Singularity
- Markdown (and structured text description)
- **Notebook environment**
- LaTeX / Overleaf
- Mental model for computation
- SSH + server coding
- UTokyo services

- You should be familiar with them
- If you don't know any of them, study on your own. There are several online resources available
- I will not cover them in detail in class.

Notebook environment

Google Colaboratory

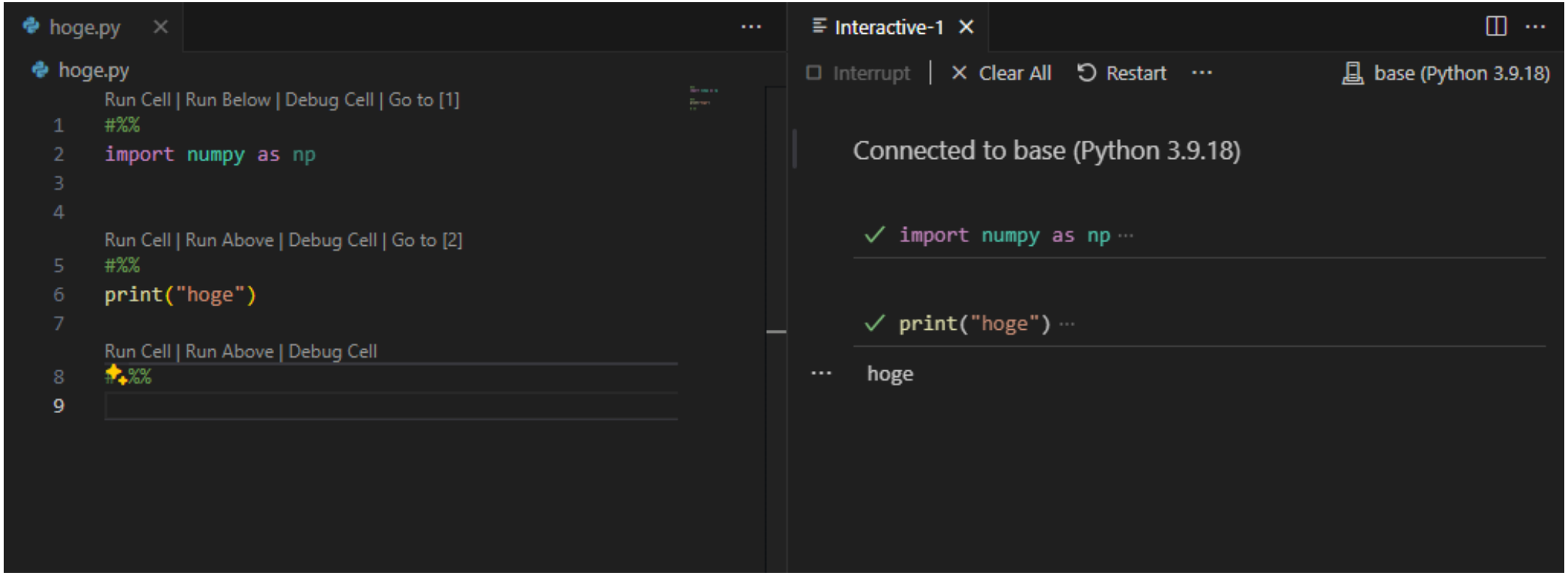


Jupyter notebook

- Some people may argue that notebooks are not useful
- However, it's beneficial to at least understand what a notebook is

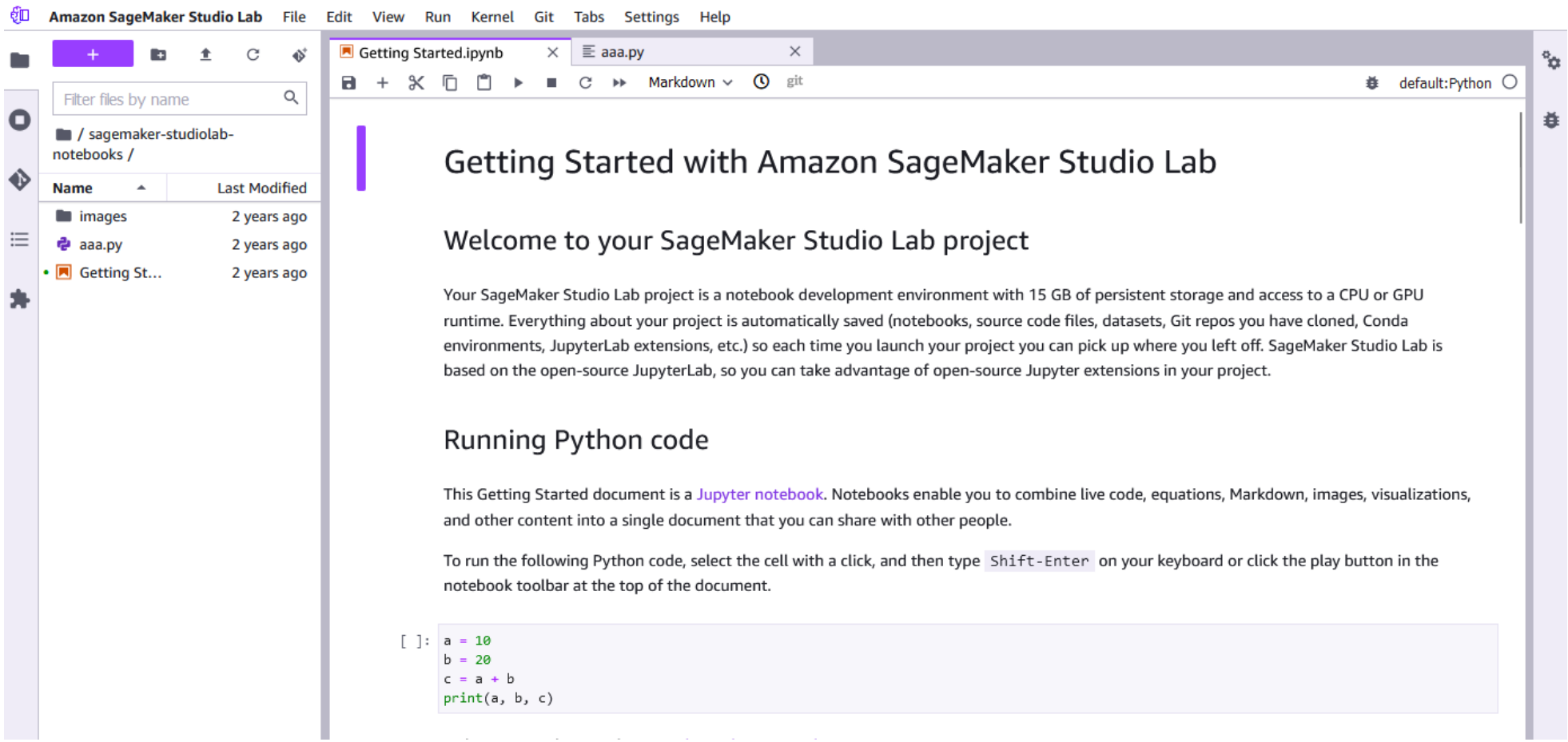
Notebook environment

- Python interactive mode: sometimes useful (not .ipynb, but .py)
- Version-controrable (for source codes only)



Notebook environment

- AWS SageMaker Studio Labs (Amazon's counterpart of Google Colab)



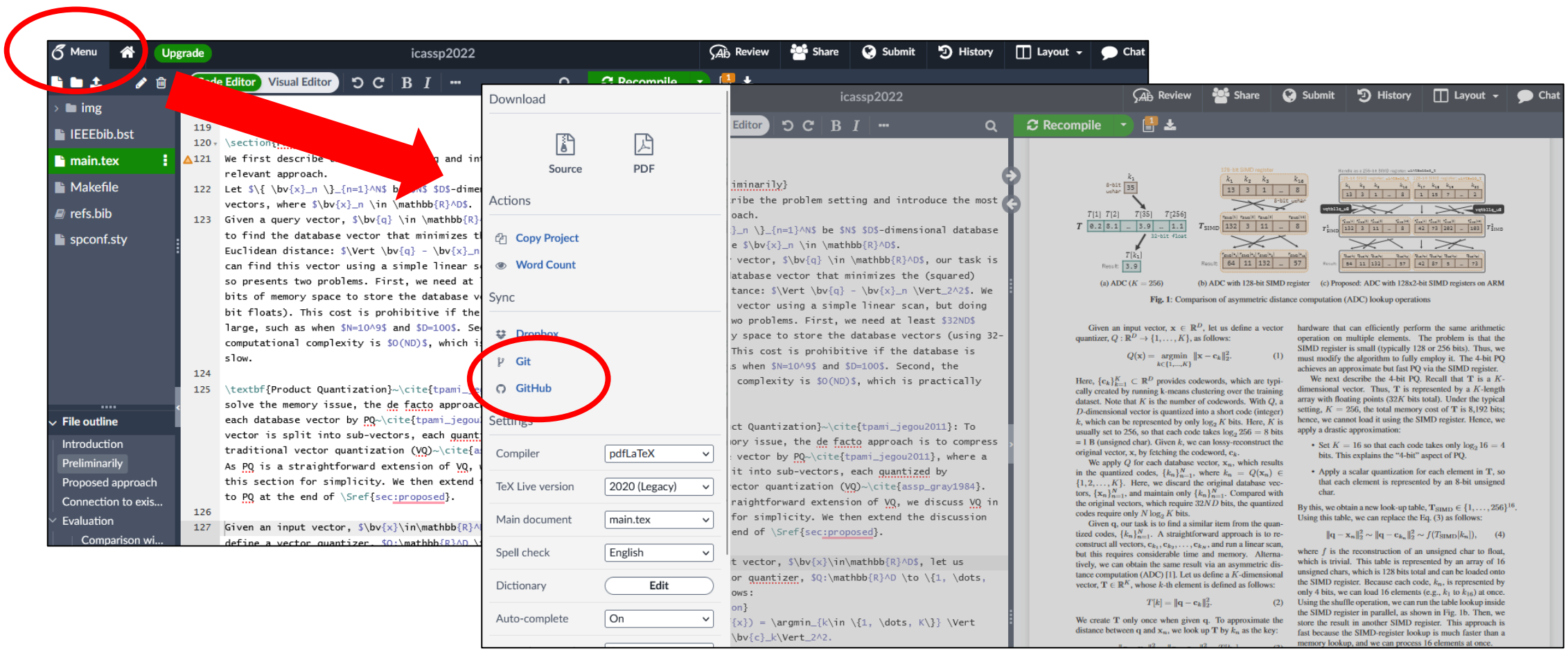
Agenda

- Git / GitHub (basic usage)
- Make
- Docker / Singularity
- Markdown (and structured text description)
- Notebook environment
- **LaTeX / Overleaf**
- Mental model for computation
- SSH + server coding
- UTokyo services

- You should be familiar with them
- If you don't know any of them, study on your own. There are several online resources available
- I will not cover them in detail in class.

LaTeX / Overleaf

- You must use LaTeX. Overleaf is a popular tool for collaborative writing
- You can back up data by git or github. You must do it!



Agenda

- Git / GitHub (basic usage)
- Make
- Docker / Singularity
- Markdown (and structured text description)
- Notebook environment
- LaTeX / Overleaf
- **Mental model for computation**
- SSH + server coding
- UTokyo services

- You should be familiar with them
- If you don't know any of them, study on your own. There are several online resources available
- I will not cover them in detail in class.

Mental model for computation

- Various mental models for computation
- It's important to carefully consider
  - ✓ which model to use
  - ✓ what tools to employ

My PC

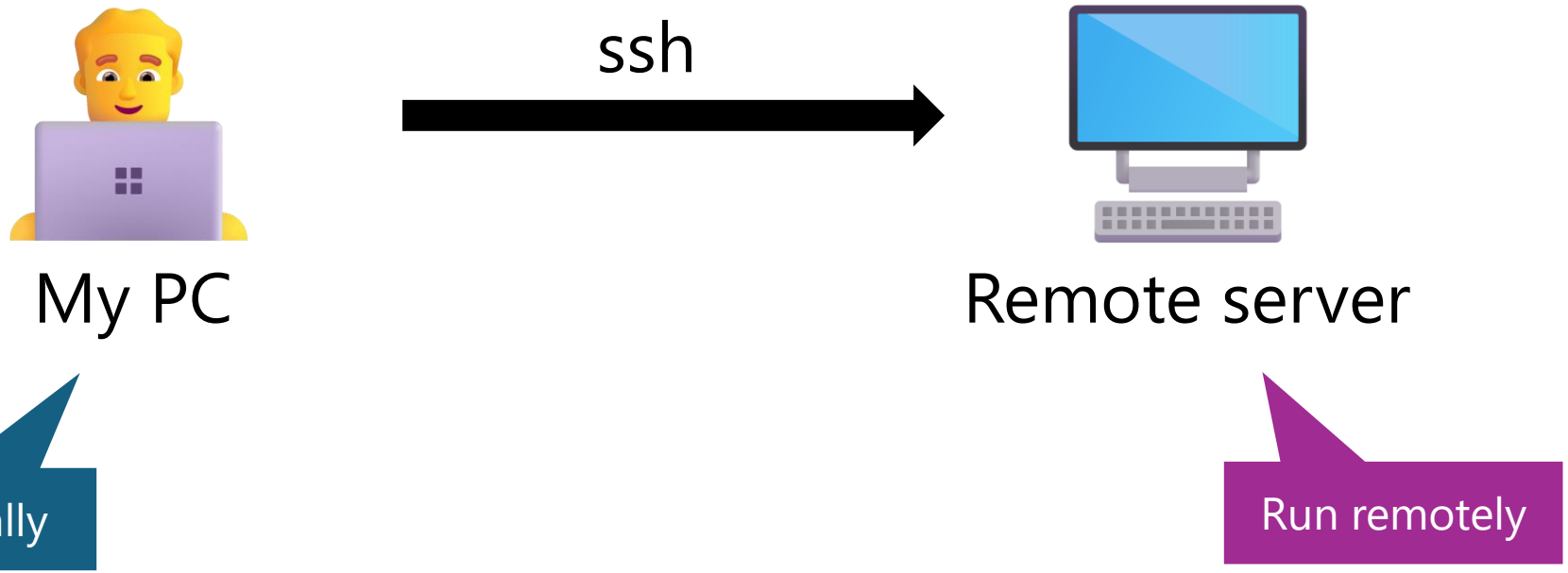
Remote server

Code locally

Run locally

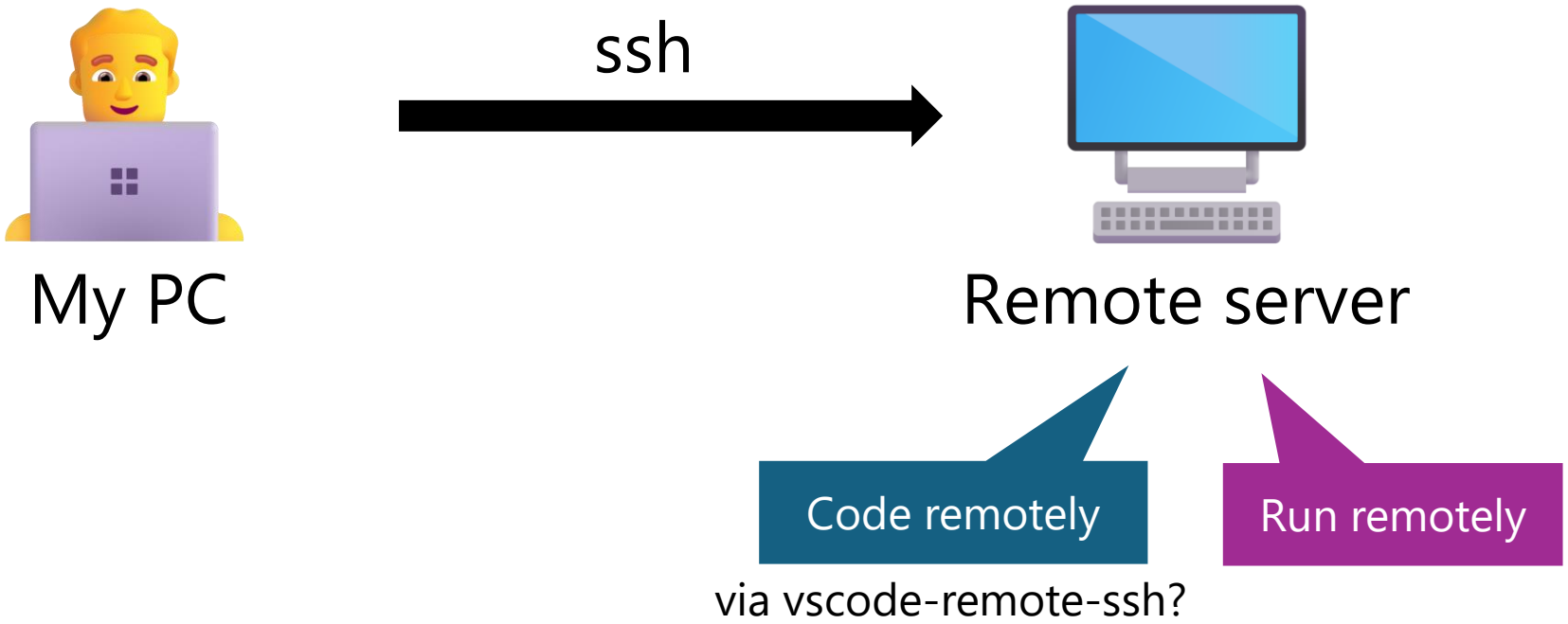
Mental model for computation

- Various mental models for computation
- It's important to carefully consider
  - ✓ which model to use
  - ✓ what tools to employ



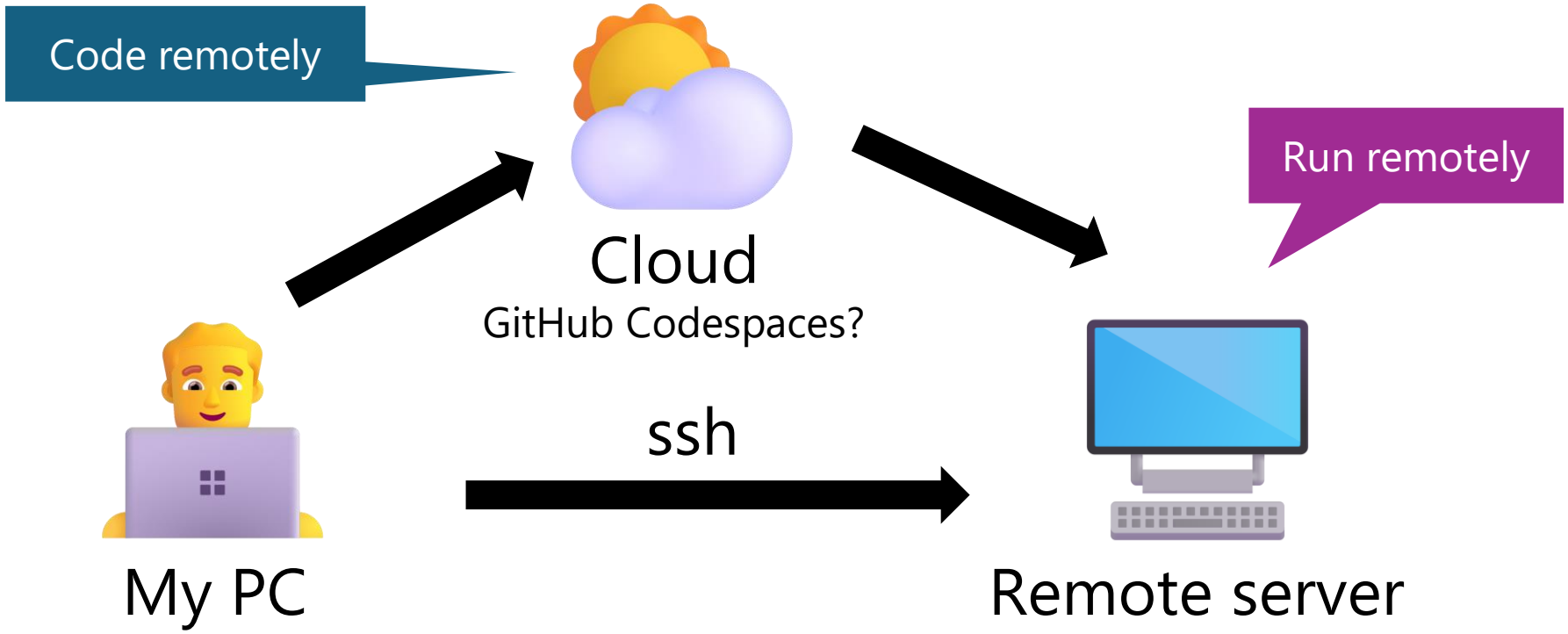
Mental model for computation

- Various mental models for computation
- It's important to carefully consider
  - ✓ which model to use
  - ✓ what tools to employ



Mental model for computation

- Various mental models for computation
- It's important to carefully consider
  - ✓ which model to use
  - ✓ what tools to employ



Mental model for computation

- Various mental models for computation
- It's important to carefully consider
  - ✓ which model to use
  - ✓ what tools to employ

My PC

ssh

Remote server

submit jobs

HPC Clusters

Code remotely

via vscode-remote-ssh?

Run remotely

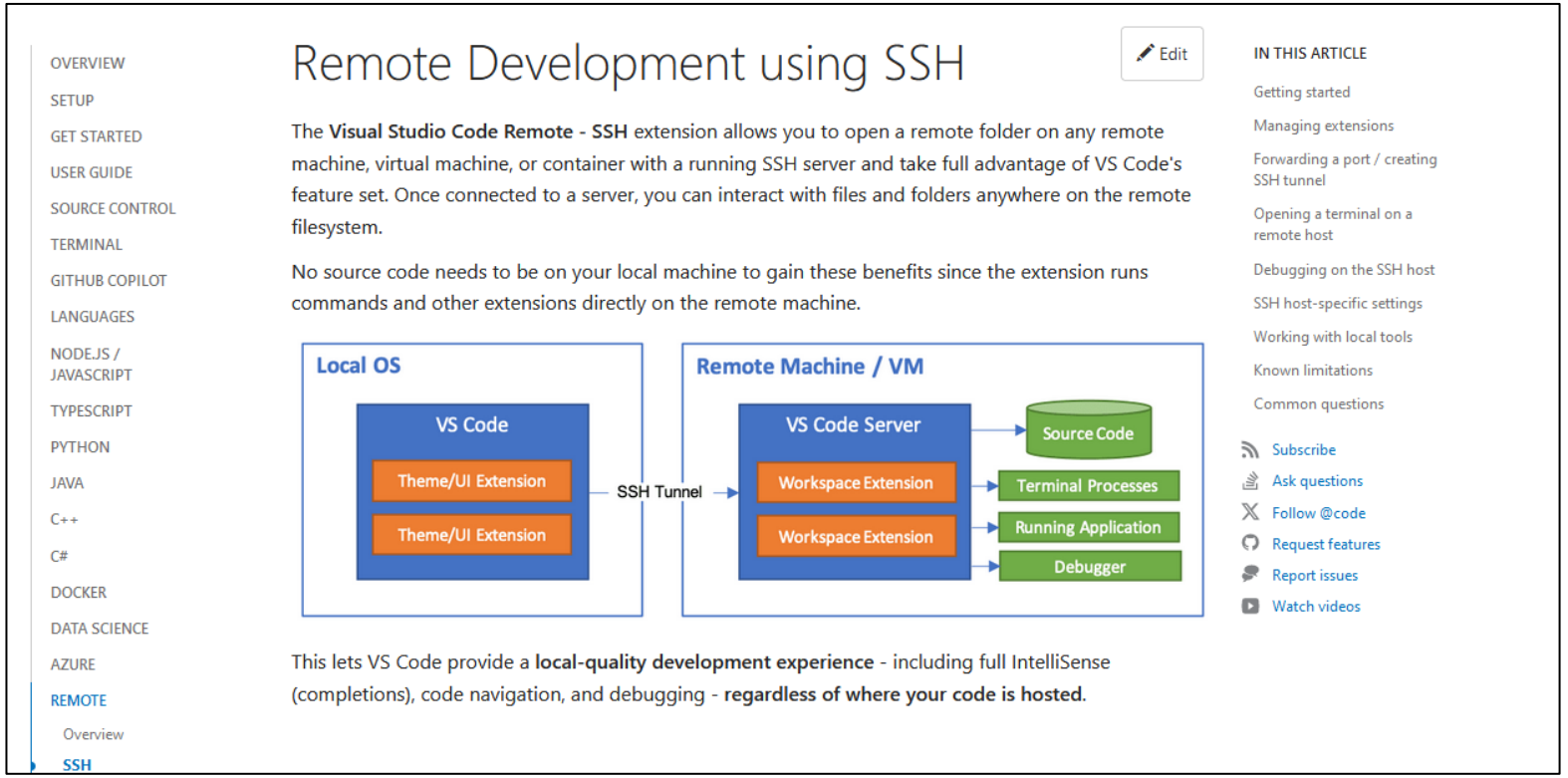
Agenda

- Git / GitHub (basic usage)
- Make
- Docker / Singularity
- Markdown (and structured text description)
- Notebook environment
- LaTeX / Overleaf
- Mental model for computation
- **SSH + server coding**
- UTokyo services

- You should be familiar with them
- If you don't know any of them, study on your own. There are several online resources available
- I will not cover them in detail in class.

SSH + server coding

- You should be able to
  - ✓ code via GUI over ssh (vscode-remote-ssh, etc)
  - ✓ ssh, then code on terminal (😎 emacs, vim, nano, etc)
- DEMO



https://code.visualstudio.com/docs/remote/ssh

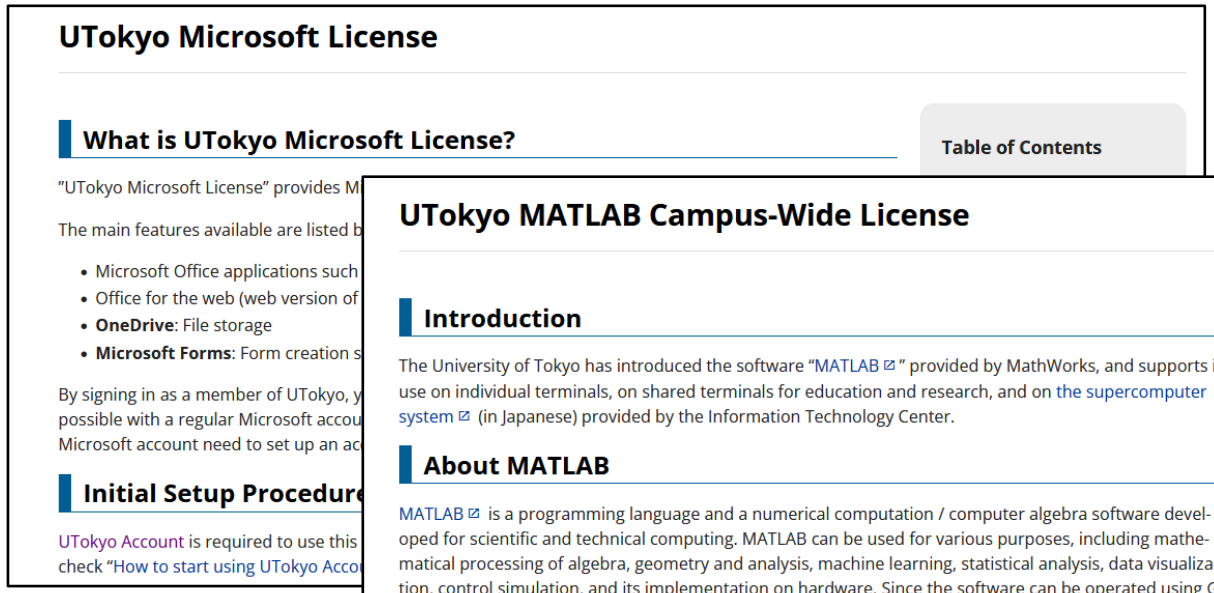
Agenda

- Git / GitHub (basic usage)
- Make
- Docker / Singularity
- Markdown (and structured text description)
- Notebook environment
- LaTeX / Overleaf
- Mental model for computation
- SSH + server coding
- **UTokyo services**

- You should be familiar with them
- If you don't know any of them, study on your own. There are several online resources available
- I will not cover them in detail in class.

UTokyo services

- Several services are available. Check them!
  - utelecon: https://utelecon.adm.u-tokyo.ac.jp/
  - ECCS: https://www.ecc.u-tokyo.ac.jp/



UTokyo Microsoft License

What is UTokyo Microsoft License?

Table of Contents

"UTokyo Microsoft License" provides M

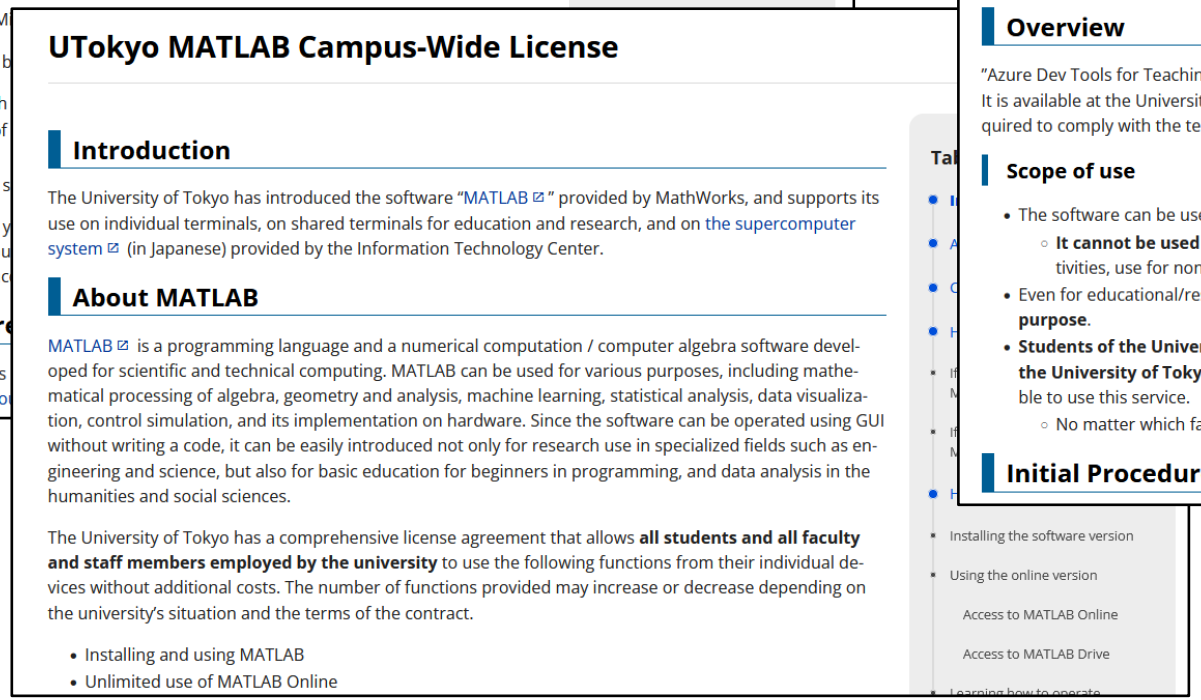
The main features available are listed b

- Microsoft Office applications such
- Office for the web (web version of
- **OneDrive**: File storage
- **Microsoft Forms**: Form creation s

By signing in as a member of UTokyo, y
possible with a regular Microsoft accou
Microsoft account need to set up an ac

Initial Setup Procedur

UTokyo Account is required to use this
check "How to start using UTokyo Acco



UTokyo MATLAB Campus-Wide License

Introduction

The University of Tokyo has introduced the software "MATLAB" provided by MathWorks, and supports its use on individual terminals, on shared terminals for education and research, and on the supercomputer system (in Japanese) provided by the Information Technology Center.

About MATLAB

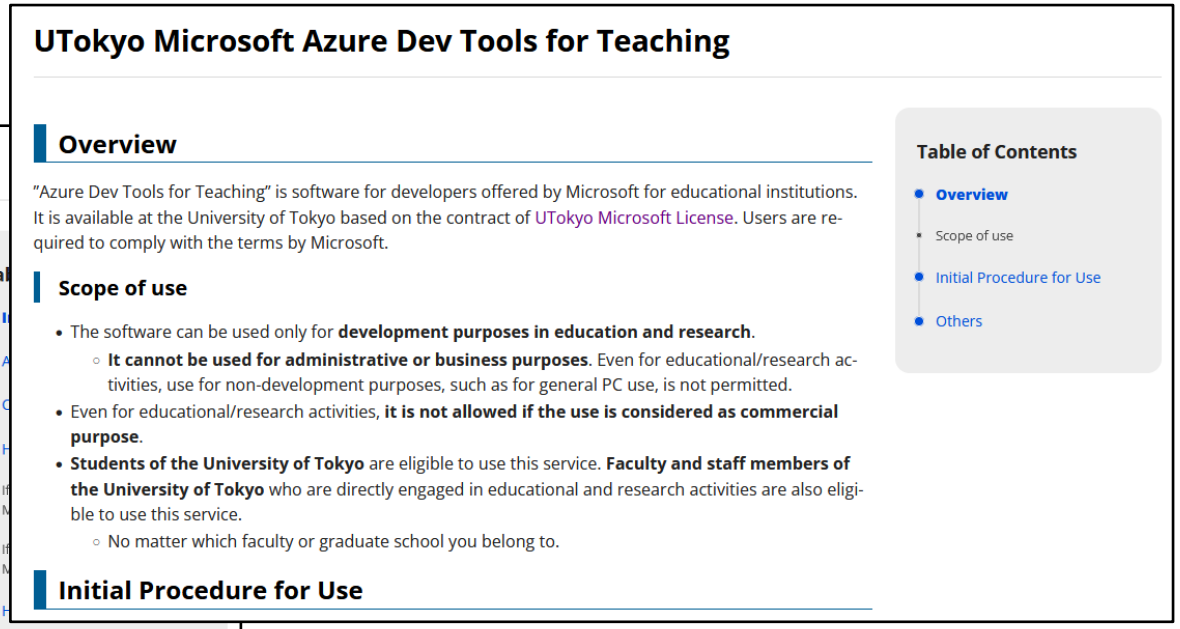
MATLAB is a programming language and a numerical computation / computer algebra software developed for scientific and technical computing. MATLAB can be used for various purposes, including mathematical processing of algebra, geometry and analysis, machine learning, statistical analysis, data visualization, control simulation, and its implementation on hardware. Since the software can be operated using GUI without writing a code, it can be easily introduced not only for research use in specialized fields such as engineering and science, but also for basic education for beginners in programming, and data analysis in the humanities and social sciences.

The University of Tokyo has a comprehensive license agreement that allows **all students and all faculty and staff members employed by the university** to use the following functions from their individual devices without additional costs. The number of functions provided may increase or decrease depending on the university's situation and the terms of the contract.

- Installing and using MATLAB
- Unlimited use of MATLAB Online

Tal

- Installing the software version
- Using the online version
  - Access to MATLAB Online
  - Access to MATLAB Drive



UTokyo Microsoft Azure Dev Tools for Teaching

Overview

"Azure Dev Tools for Teaching" is software for developers offered by Microsoft for educational institutions. It is available at the University of Tokyo based on the contract of UTokyo Microsoft License. Users are required to comply with the terms by Microsoft.

Scope of use

- The software can be used only for **development purposes in education and research**.
  - **It cannot be used for administrative or business purposes**. Even for educational/research activities, use for non-development purposes, such as for general PC use, is not permitted.
- Even for educational/research activities, **it is not allowed if the use is considered as commercial purpose**.
- **Students of the University of Tokyo** are eligible to use this service. **Faculty and staff members of the University of Tokyo** who are directly engaged in educational and research activities are also eligible to use this service.
  - No matter which faculty or graduate school you belong to.

Initial Procedure for Use

Table of Contents

- **Overview**
- Scope of use
- Initial Procedure for Use
- Others

Schedule

Date (2024)	Contents	Presented by
~~Week 1, Apr 10~~	~~Introduction. Review of fundamental concepts~~	~~Yusuke, Koya, Yuki, Jun~~
Week 2, Apr 17	Equations and pseudo-codes	Yusuke Matsui
Week 3, Apr 24	Presentation	Koya Narumi
Week 4, May 1	Tables and plots	Yusuke Matsui
Week 5, May 8	Figures	Koya Narumi
Week 6, May 22	Videos	Koya Narumi
Week 7, May 29	Invited Talk 1	Dr. Yoshiaki Bando (AIST)
Week 8, June 5	Invited Talk 2	Prof. Katie Seaborn (Tokyo Tech)
Week 9, June 12	GitHub in depth	Yusuke Matsui
Week 10, June 19	Automation of research and research dissemination (Web, Cloud, CI/CD)	Jun Kato
Week 11, June 26	Research community	Jun Kato
Week 12, July 3	3DCG illustrations	Yuki Koyama
Week 13, July 10	Final presentations	-