Introduction

Programming Language Theory

A Few Things

- Additional Enrollments are possible during add/drop period.
- I’m not sure about how many spots will be opened.
- Please complete your survey in e-Class.
- Any feedback is welcome.

Syllabus

- Programming Language Theory
- Pre-requisites
  - Good understanding of at least one programming language.
  - Basic understanding of C++ and Java.
- Contents
  - Programming Language Design Principles
  - Programming Language Concepts
  - Programming Language Paradigm
  - A few programming languages in different paradigms.

Course Organization

- We will first look at programming language design principles and concepts for the first half of the semester.
- During the first half, there will be more lectures and less practices.
- For the next half, we will study various programming language paradigms with selected specific languages.
- Then we will have more practices to learn these various programming languages.

Syllabus

- Assessment
  - Midterm Exam 30%
  - Final Exam 40%
  - Assignments **30%** → Sorry for the typo.
  - No late submission allowed.
  - No team project!? Maybe group assignments?

Assignments

- For the first half of the semester, you’re required to submit some assignments (maybe 1~2) to solve some problems.
- For the last half, you will need to write down small programs in different languages we will learn.
- Simple tasks for verifying attendance won’t be counted as assignments.
- Group Assignments: Since there will be more enrollments, I’m considering group assignments at the latter part of semester.

Harsh Truth

- The most annoying thing in team projects is that only some of the team work hard.

- Agrawal, Amritanshu, et al. ***We don't need another hero? the impact of "heroes" on software development.*** ICSE 2018: Software Engineering in Practice.

- A project has "Hero Developers" when 80% of contributions are delivered by 20% of the developers.

- As projects grow in size, nearly all projects become hero projects.

- In Enterprise projects, heroes increase the rate of completing program enhancement.

Syllabus

- No official textbooks.
  - If you want to study further, we may check the following books as references.
  - Michael L. Scott, Programming Language Pragmatics, 4th Edition, Morgan Kaufmann
  - Maurizio Gabbrielli and Simone Martini, Programming Languages: Principles and Paradigms, Springer-Verlag London
- Please, Don’t buy the books unless you’re really into it.

Syllabus

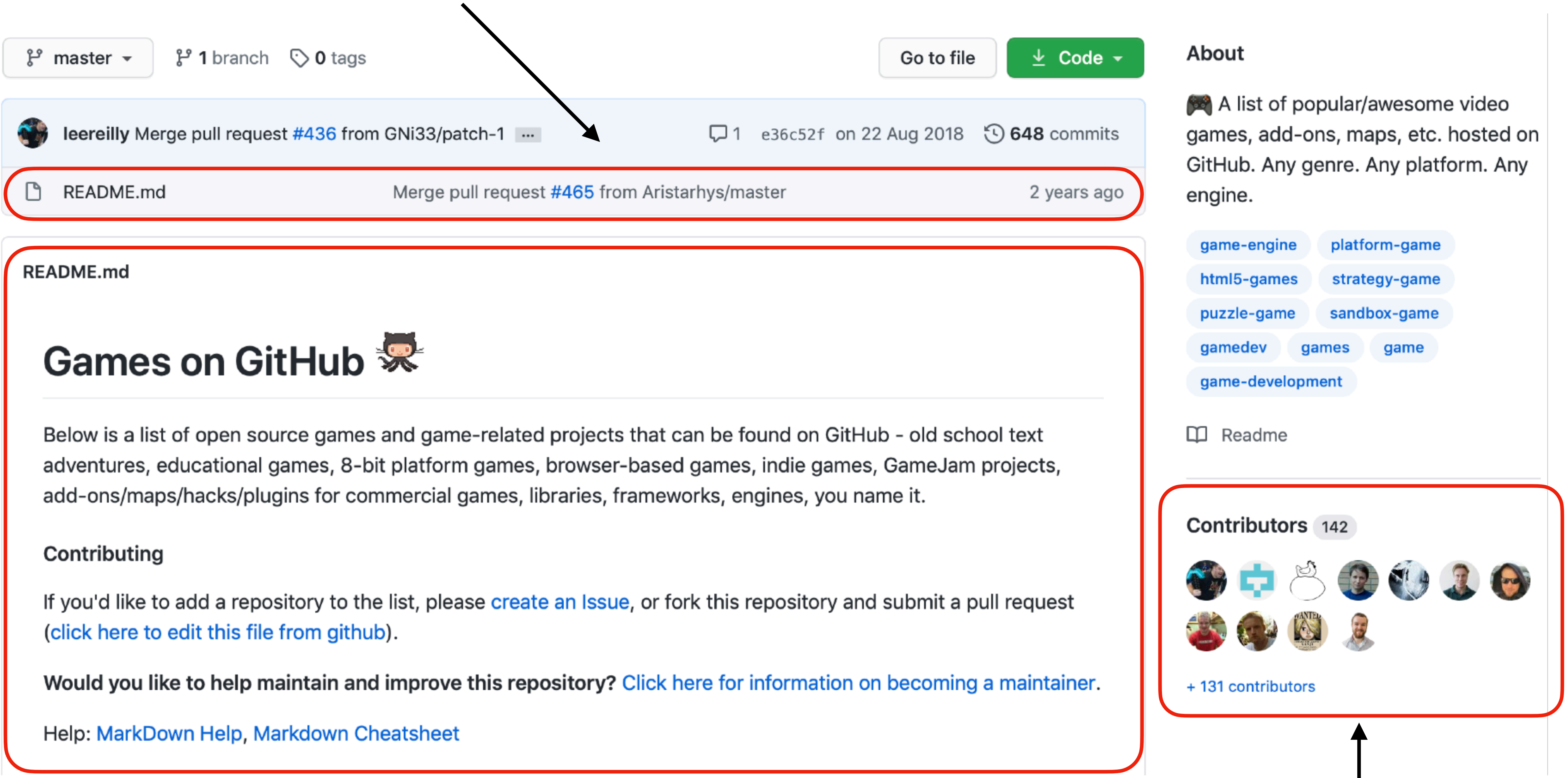
- Communications
  - Jindae Kim (김진대), Mirae Hall(미래관) 331
  - Office Hour: Anytime, but maybe not in the office or during a meeting. Recommend you to check via email / e-Class.
  - E-mail: jindae.kim@seoultech.ac.kr
  - Using GitHub

GitHub

- One of the most popular, flourishing Software Project Hosting Services.
- More than 125 million projects are being hosted by GitHub.
- Supporting source code management with Git.
- Also provides issue management system.
- It is more likely that you will also use GitHub for development after graduate.
- It’s not just for software projects, but for public archives or discussions.

An Example

Where files of a project are shown



leereilly Merge pull request #436 from GNi33/patch-1 … 1 e36c52f on 22 Aug 2018 648 commits

README.md Merge pull request #465 from Aristarhys/master 2 years ago

README.md

Games on GitHub

Below is a list of open source games and game-related projects that can be found on GitHub - old school text adventures, educational games, 8-bit platform games, browser-based games, indie games, GameJam projects, add-ons/maps/hacks/plugins for commercial games, libraries, frameworks, engines, you name it.

Contributing

If you'd like to add a repository to the list, please create an Issue, or fork this repository and submit a pull request (click here to edit this file from github).

Would you like to help maintain and improve this repository? Click here for information on becoming a maintainer.

Help: MarkDown Help, Markdown Cheatsheet

About

🎮 A list of popular/awesome video games, add-ons, maps, etc. hosted on GitHub. Any genre. Any platform. Any engine.

game-engine platform-game html5-games strategy-game puzzle-game sandbox-game gamedev games game game-development

Readme

Contributors 142

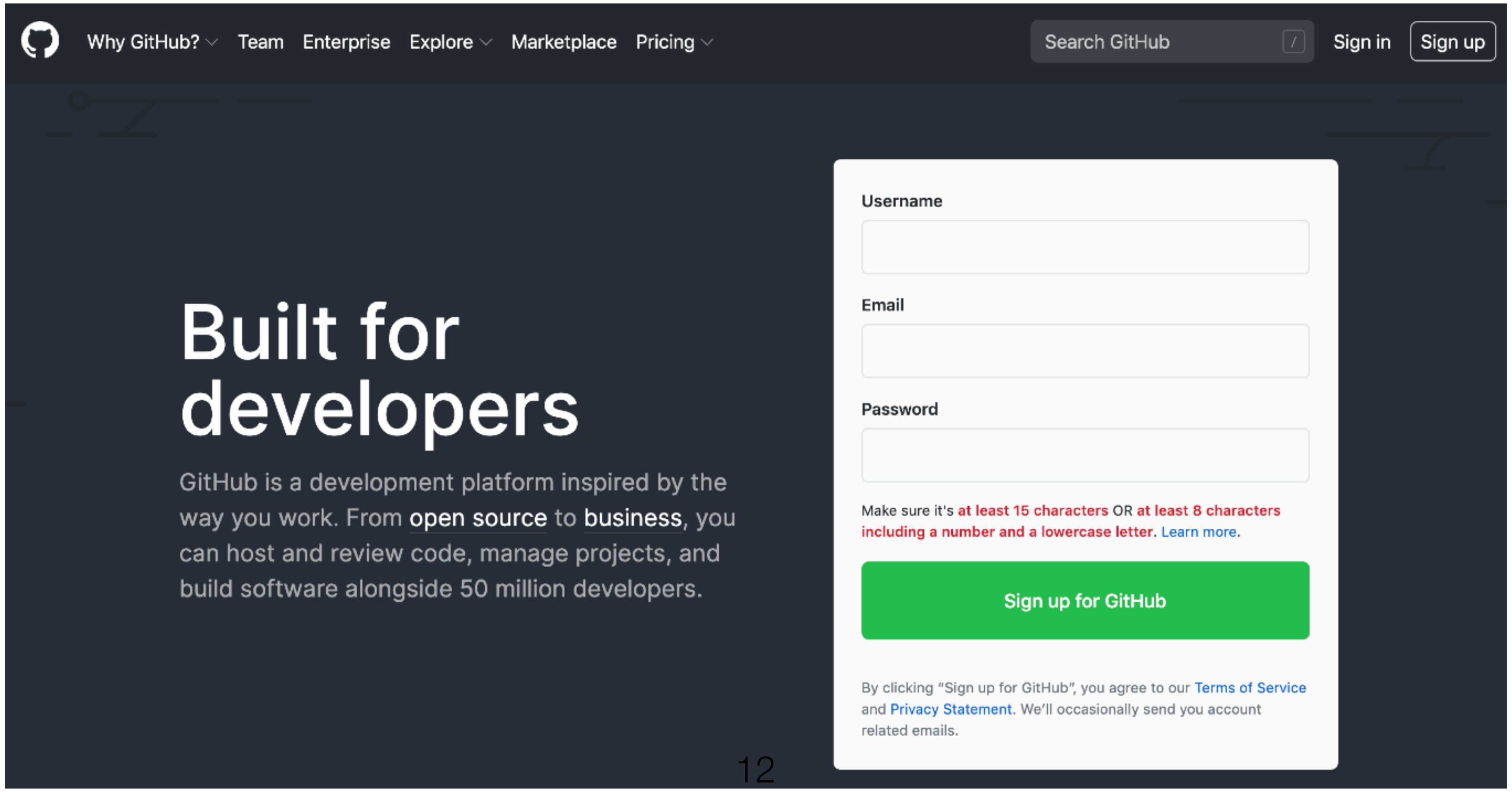
+ 131 contributors

Content of README.md File

Participants of a Project

Creating GitHub ID

- https://github.com/
- Create Profile + Repository → Using GitHub page link as a part of your career portfolio
- Typing Username (ID), Email, Password and it’s done!



Built for developers

GitHub is a development platform inspired by the way you work. From open source to business, you can host and review code, manage projects, and build software alongside 50 million developers.

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.

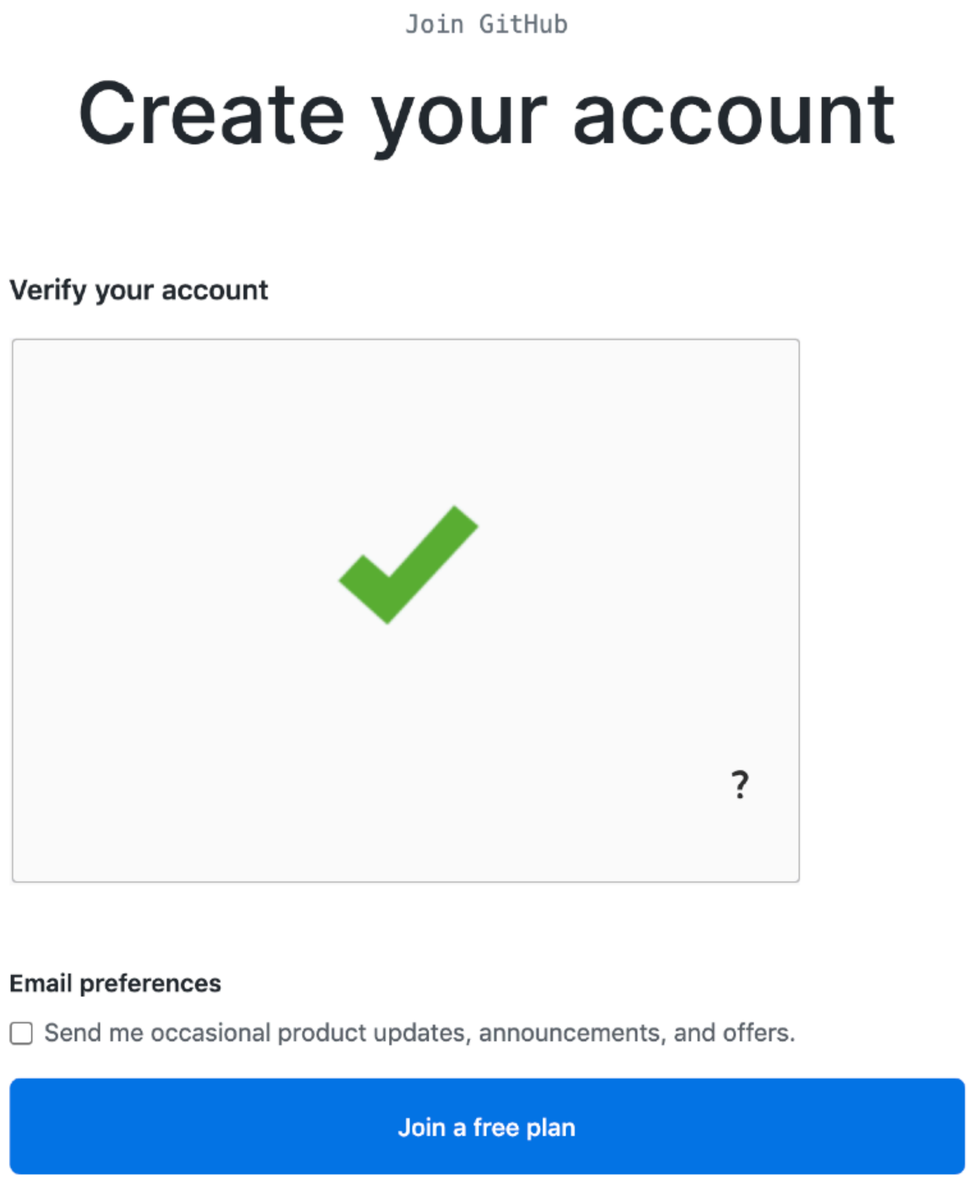
By clicking “Sign up for GitHub”, you agree to our Terms of Service and Privacy Statement. We’ll occasionally send you account related emails.

Creating GitHub ID

- Solve a simple puzzle to verify, then ‘Join a free plan’ button becomes active.

- After that, a simple survey which you can ignore.

- Lastly, email verification → go to your email account and proceed with verification.



Join GitHub

Create your account

Verify your account

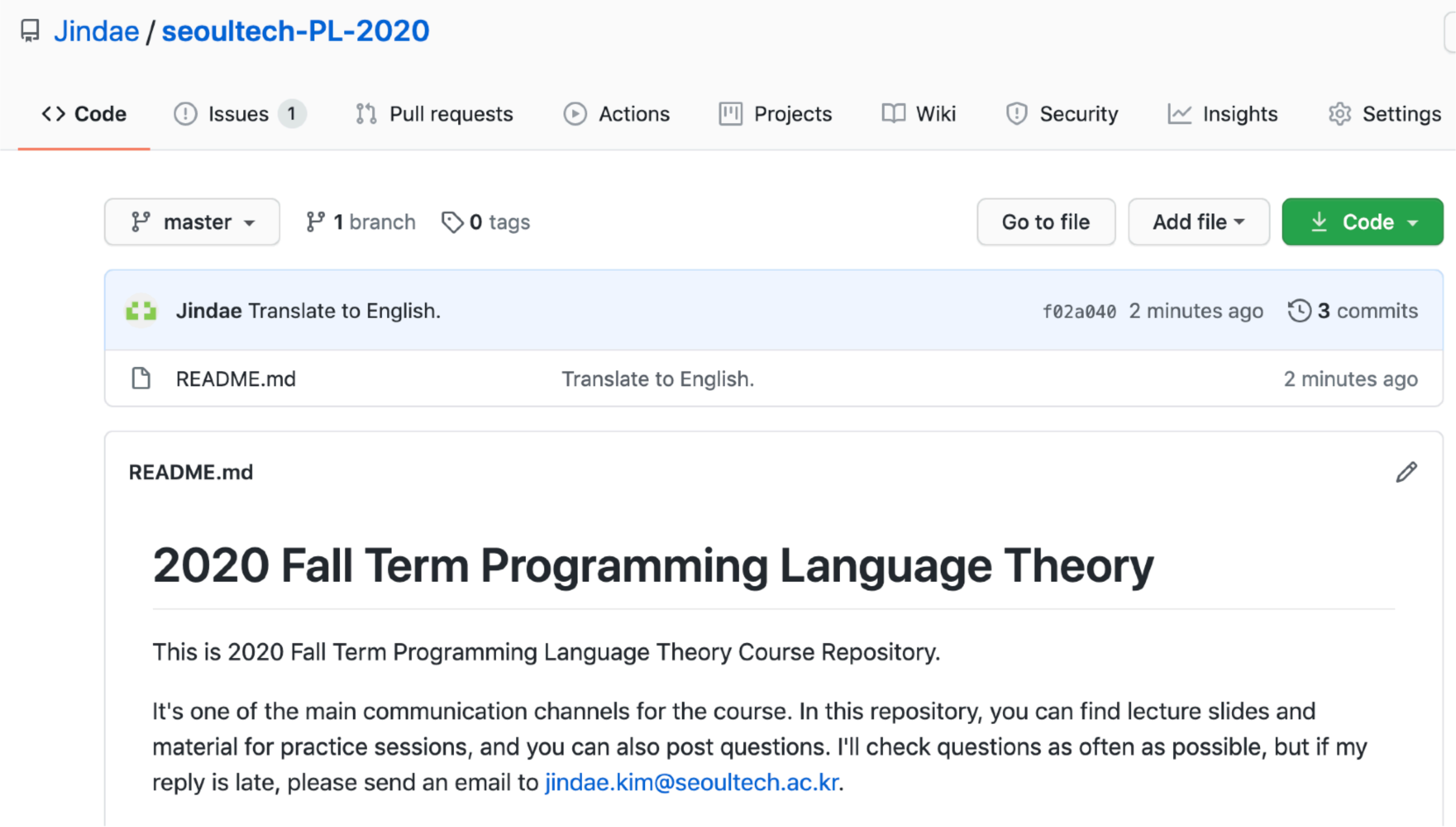
?

Email preferences

□ Send me occasional product updates, announcements, and offers.

Here is what ours looks like

https://github.com/Jindae/seoultech-PL-2020



Jindae Translate to English. f02a040 2 minutes ago 3 commits

README.md Translate to English. 2 minutes ago

README.md

2020 Fall Term Programming Language Theory

This is 2020 Fall Term Programming Language Theory Course Repository.

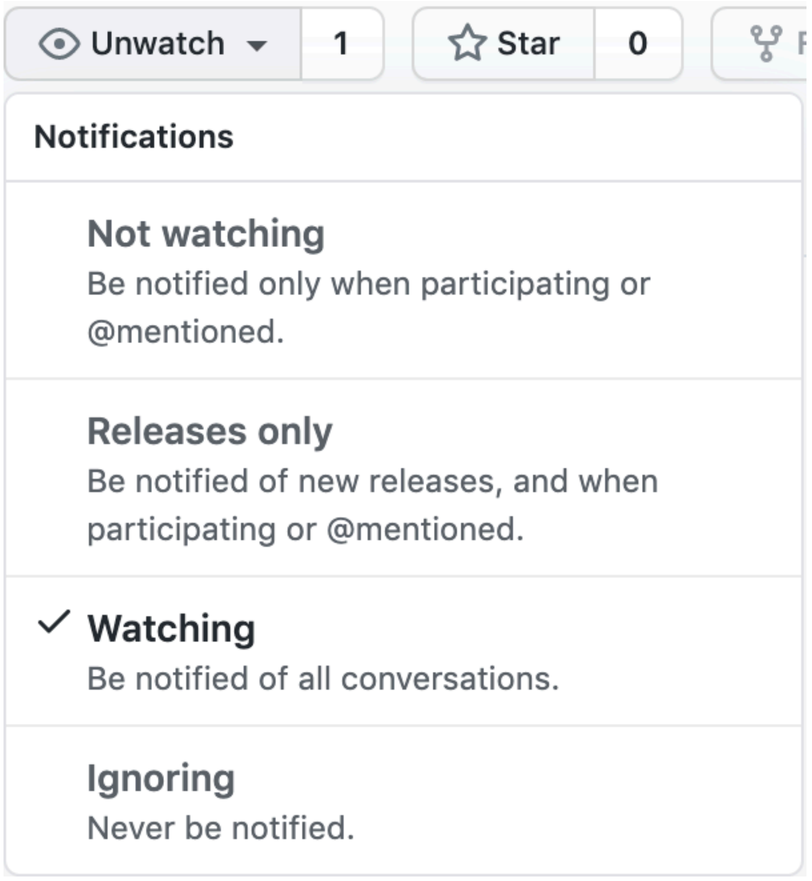
It's one of the main communication channels for the course. In this repository, you can find lecture slides and material for practice sessions, and you can also post questions. I'll check questions as often as possible, but if my reply is late, please send an email to jindae.kim@seoultech.ac.kr.

What’s on Course GitHub?

- Lecture Slides like this one.
- Material for Practice Sessions
- Files for Assignments
- Q & A
- Other material regarding this course.

Don’t Forget to Watch!

- When there is any change in the repository, you can get a notification if you change your state as ‘**Watching**’ or ‘**Releases only**’.
- If there are too many unwanted notifications, you can change your state to ‘**Not Watching**’, but *still you can get notifications for your questions or your name’s mentioned*.



Notifications

Not watching
Be notified only when participating or @mentioned.

Releases only
Be notified of new releases, and when participating or @mentioned.

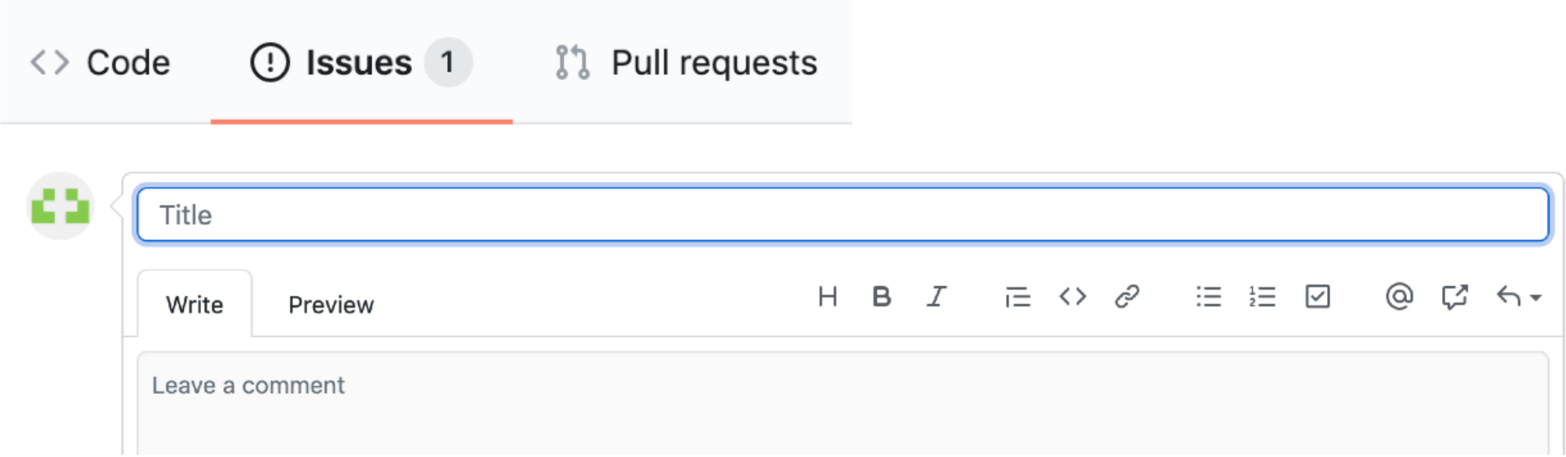
✓ **Watching**
Be notified of all conversations.

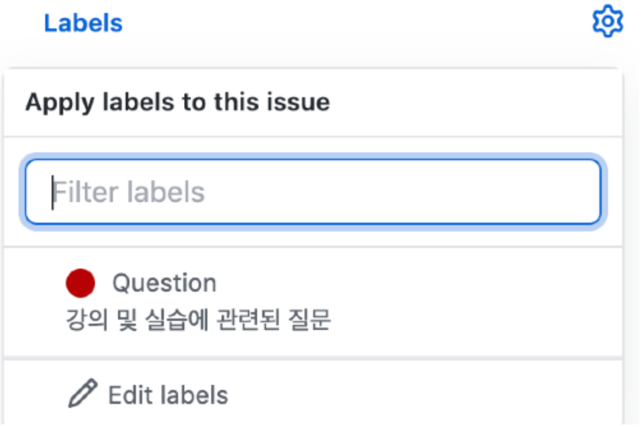
Ignoring
Never be notified.

Posting Enquiries

- You can ask questions about the course by posting GitHub Issues.
- Or you can email me - Please put [SeoulTech-PL] on the subject of your email. If not, your email may filtered as spam or reply could be getting late.

How to Post an Issue in GitHub?





- Go to Issues > New Issue, then you can see the issue posting window.

- Put title and content to post your question.

- Don’t forget to select Question label on the right!

- For more details, please refer to course GitHub.

Programming Language Theory

- So far, you're mostly 'using' programming languages for software development.

- How about 'making' programming languages?

- Programming Language (PL) theory is about **how to design good programming languages**, and build a basis for programming language development.

Scope of This Course

- The aim of this course is to ***understand PL concepts and paradigms***, and use that knowledge ***to help learning new programming languages***.

- Normally PL courses cover very serious theoretical stuff from the foundation.

- However, not everyone is interested in programming language development.

- Apologizes to students who want to create their own programming languages; this course doesn't cover full.

Scope of This Course

- Still, this is actually a theory course.
- We cannot avoid studies on theoretical foundation of programming languages.
- This will help you have more deep understanding in programming languages.
- Also, many of these contents will make you look very professional!

Why PL Theory?

- There have been so many different programming languages.
- Useful common concepts among these languages have been studied, evolved, and reflected on new programming languages.
- How can we include useful concepts in a new programming language?
  - While minimize accompanying drawbacks?

PL Concepts and Paradigms

- Many programming languages are different implementations of the similar concepts following the similar paradigms.

- For example, consider a sorting program.

- You may write many sorting programs in different languages implementing different algorithms.

- Still, they’re sorting programs which place something in order.

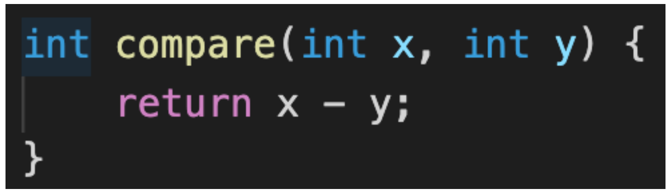
PL Concepts and Paradigms

- **PL concepts**: more like individual features.
- e.g.) data types, control flow, expression, statements, variables, functions, etc.
- **PL Paradigms**: principles and strategies which a PL follows.
- e.g.) Procedural, Imperative, Object oriented, Functional, Logic, etc.

PL Concepts and Paradigms

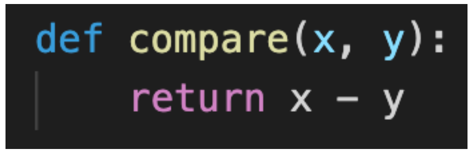
- PLs share common concepts and paradigms.
- Once you understand those concepts and paradigms, learning a PL is now learning how the PL implements them (e.g., syntax).

C/C++



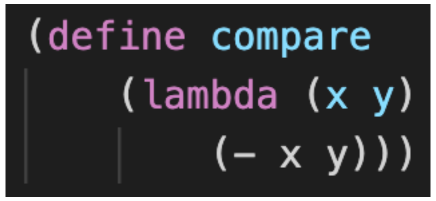
```c
int compare(int x, int y) {
    return x - y;
}
```

Python



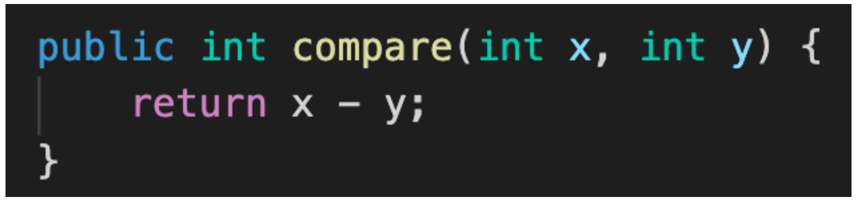
```python
def compare(x, y):
    return x - y
```

Scheme



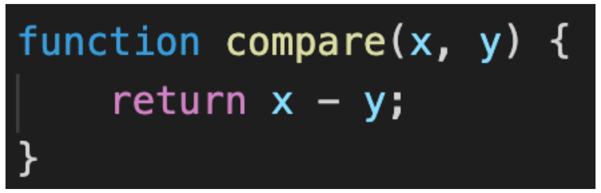
```scheme
(define compare
    (lambda (x y)
        (- x y)))
```

Java



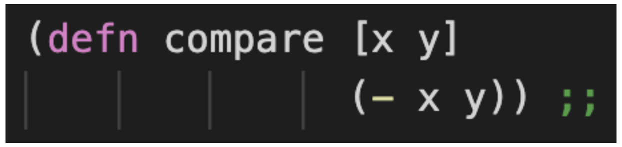
```java
public int compare(int x, int y) {
    return x - y;
}
```

JavaScript



```javascript
function compare(x, y) {
    return x - y;
}
```

Clojure



```clojure
(defn compare [x y]
        (- x y)) ;;
```

Summary

- Syllabus
- Scope and Organization of this course.
- PL Concepts and Paradigms.