



**TRƯỜNG ĐẠI HỌC VINH**  
**VIỆN KỸ THUẬT VÀ CÔNG NGHỆ**

**TÓM TẮT BÀI GIẢNG**  
**KỸ THUẬT LẬP TRÌNH**  
**(TÀI LIỆU LƯU HÀNH NỘI BỘ)**

**Đơn vị: Viện Kỹ thuật và Công nghệ**

**Nghe An, 2022**

# THÔNG TIN HỌC PHẦN

- Tên học phần: **Kỹ thuật lập trình**
- Mã học phần: **ELE20004**
- Khối kiến thức: Kiến thức cơ sở ngành
- Số tín chỉ: 04
  - Lý thuyết: 15 tiết; Thực hành: 30 tiết Đồ án: 15 tiết; Tự học: 150 tiết
- Vị trí học phần:
  - Học phần tiên quyết: Tin học nhóm ngành kỹ thuật (INF20004)
  - Học phần song hành: Không

# MÔ TẢ HỌC PHẦN

Kỹ thuật lập trình là học phần đầu tiên về lập trình, dành cho các sinh viên chưa có kiến thức nào về lập trình trước đó.

Học phần cung cấp những kiến thức và kỹ năng căn bản về lập trình bao gồm hai phương pháp lập trình: lập trình có cấu trúc và lập trình hướng đối tượng. Bên cạnh đó sinh viên còn được củng cố về các kỹ năng về làm việc nhóm để viết chương trình phần mềm giải quyết các vấn đề trong thực tiễn.

# MỤC TIÊU HỌC PHẦN

- **G1** - Sử dụng các thuật toán để giải quyết vấn đề;
- **G2** - Thực hiện lập trình bằng ngôn ngữ Python;
- **G3** - Nắm vững kiến thức về lập trình hướng đối tượng;
- **G4** - Thể hiện phong cách lập trình chuyên nghiệp;
- **G5** - Thực hiện giải quyết vấn đề bằng lập trình phần mềm;
- **G6** - Thể hiện kỹ năng làm việc nhóm và giao tiếp hiệu quả.

# NỘI DUNG HỌC PHẦN

- **CHƯƠNG 1** - Giới thiệu về máy tính và lập trình
- **CHƯƠNG 2** - Ngôn ngữ lập trình Python
- **CHƯƠNG 3** - Lập trình hàm
- **CHƯƠNG 4** - Các kiểu dữ liệu có cấu trúc
- **CHƯƠNG 5** - Thiết kế module
- **CHƯƠNG 6** - Lập trình hướng đối tượng
- **CHƯƠNG 7** - Thao tác trên tập tin và thư mục
- **CHƯƠNG 8** - Lập trình giao diện

# HÌNH THỨC ĐÁNH GIÁ

▪ Đánh giá quá trình:	<b>50%</b>
▪ Ý thức học tập	10%
▪ Hồ sơ học tập (bài tập)	20%
▪ Đánh giá giữa kỳ (thực hành)	20%
▪ Đánh giá cuối kỳ:	<b>50%</b>
▪ Đồ án	50%

# NGUỒN HỌC LIỆU

## **Giáo trình:**

[1]. Allen B. Downey, Think Python, O'Reilly Media, Inc, 2015.

[2]. Võ Duy Tuấn, Python cơ bản, Ebook, 2016.

## **Tài liệu tham khảo:**

[3]. Magnus Lie Hetland, Beginning Python: From Novice to Professional, APress, 2008.

[4]. Website: <https://docs.python.org/3.7/tutorial>.

# QUY ĐỊNH HỌC PHẦN

- Sinh viên phải tham dự tối thiểu 80% thời gian lên lớp;
- Có đầy đủ học liệu theo yêu cầu của giảng viên;
- Tham gia tích cực hoạt động thảo luận theo nhóm;
- Báo cáo bài tập trên hệ thống Github đảm bảo chất lượng, đáp ứng đầy đủ nội dung yêu cầu của môn học.
- Thực hiện đồ án (project) và báo cáo đầy đủ nội dung theo quy định.





**TRƯỜNG ĐẠI HỌC VINH**  
**VIỆN KỸ THUẬT VÀ CÔNG NGHỆ**

# **GIỚI THIỆU VỀ GIT VÀ HƯỚNG DẪN SỬ DỤNG HỆ THỐNG GITHUB**

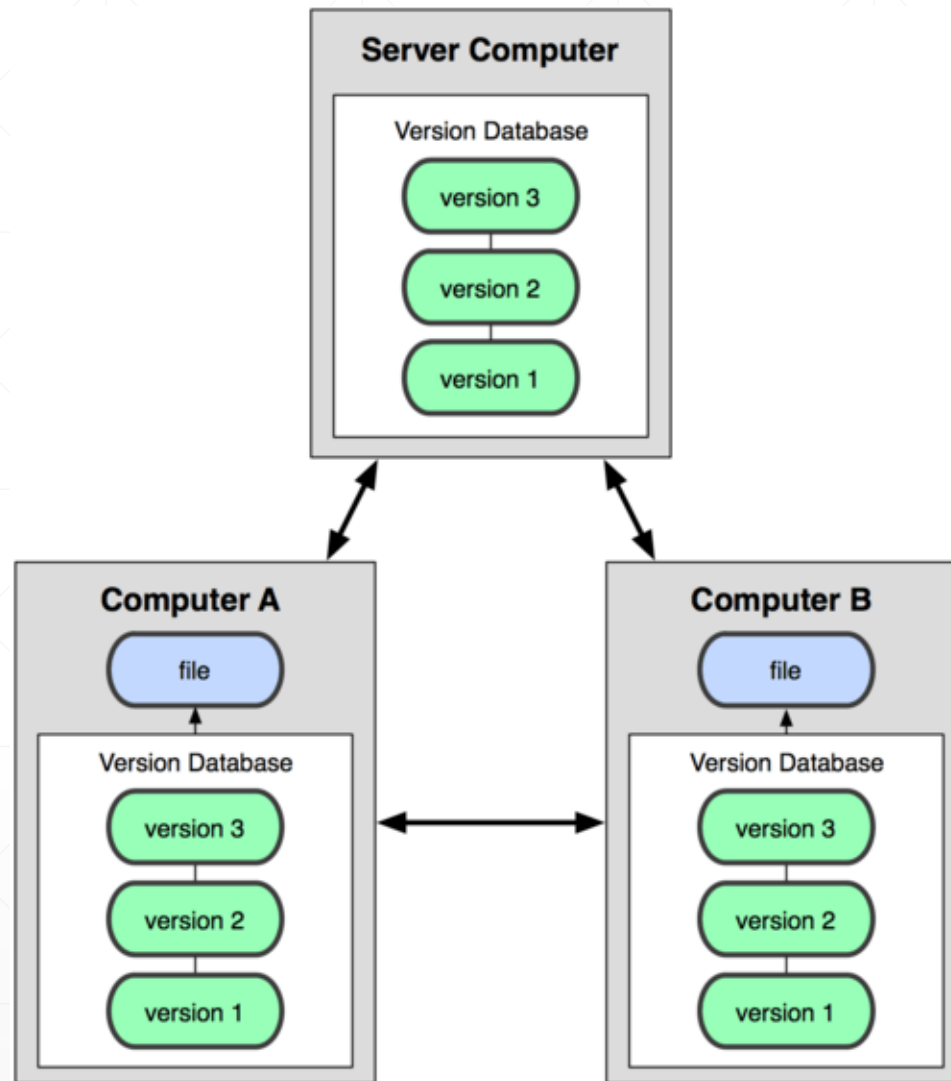
**Nghệ An, 2022**

# GIỚI THIỆU VỀ GIT

- Git là tên gọi của một Hệ thống quản lý phiên bản phân tán (Distributed Version Control System – DVCS) là một trong những hệ thống quản lý phiên bản phân tán phổ biến nhất hiện nay.
- Hệ thống giúp mỗi máy tính có thể lưu trữ nhiều phiên bản khác nhau của một mã nguồn được nhân bản (clone) từ một kho chứa mã nguồn (repository);

# GIỚI THIỆU VỀ GIT

- Mỗi thay đổi vào mã nguồn trên máy tính sẽ có thể ủy thác (commit) rồi đưa lên máy chủ nơi đặt kho chứa chính.
- Và một máy tính khác (nếu họ có quyền truy cập) cũng có thể clone lại mã nguồn từ kho chứa hoặc clone lại một tập hợp các thay đổi mới nhất trên máy tính kia.
- Trong Git, thư mục làm việc trên máy tính gọi là Working Tree.



Mô hình hoạt động của Distributed Version Control System – DVCS

# GIỚI THIỆU VỀ GIT

- **Hiểu một cách đơn giản:**

- Git sẽ giúp bạn lưu lại các phiên bản của những lần thay đổi vào mã nguồn và có thể dễ dàng khôi phục lại dễ dàng mà không cần copy lại mã nguồn rồi cất vào đâu đó.
- Một người khác có thể xem các thay đổi của bạn ở từng phiên bản, họ cũng có thể đối chiếu các thay đổi của bạn rồi gộp phiên bản của bạn vào phiên bản của họ.
- Cuối cùng là tất cả có thể đưa các thay đổi vào mã nguồn của mình lên một kho chứa mã nguồn.

# GIỚI THIỆU VỀ GITHUB

- GitHub là một dịch vụ cung cấp kho lưu trữ mã nguồn Git dựa trên nền web cho các dự án phát triển phần mềm.
- GitHub cung cấp cả phiên bản trả tiền lẫn miễn phí cho các tài khoản. Các dự án mã nguồn mở sẽ được cung cấp kho lưu trữ miễn phí.
- Tính đến tháng 4 năm 2016, GitHub có hơn 14 triệu người sử dụng với hơn 35 triệu kho mã nguồn, làm cho nó trở thành máy chủ chứa mã nguồn lớn trên thế giới.

# GIỚI THIỆU VỀ GITHUB

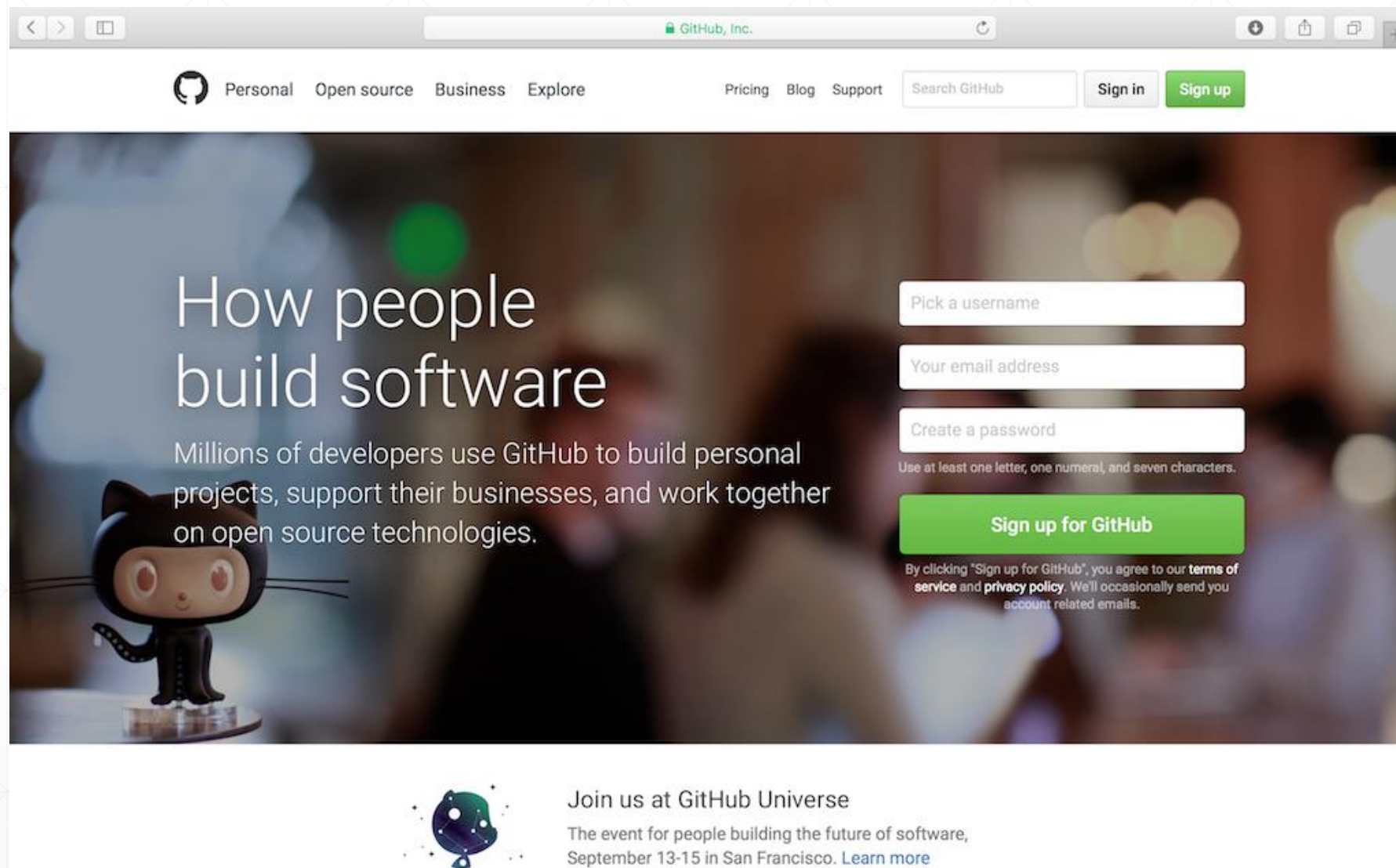
- Github đã trở thành một yếu tố có sức ảnh hưởng trong cộng đồng phát triển mã nguồn mở.
- Nhiều nhà phát triển đã bắt đầu xem nó là một sự thay thế cho sơ yếu lý lịch và một số nhà tuyển dụng yêu cầu các ứng viên cung cấp một liên kết đến tài khoản Github để đánh giá ứng viên.
- Vào ngày 4 tháng 6 năm 2018, Microsoft đã thông báo việc đạt được thỏa thuận mua lại GitHub với giá 7.5 tỷ Đôla Mỹ.

# HƯỚNG DẪN SỬ DỤNG GITHUB

- **Tạo tài khoản**

- Để bạn có thể lưu trữ mã nguồn và sử dụng các dịch vụ của GitHub, trước tiên, chúng ta phải tạo một tài khoản của GitHub.
- Để tạo tài khoản, đơn giản, trên trang web chính của GitHub, bạn bấm nút ***Sign up*** hoặc ***Sign up for GitHub***.





## Tài khoản của GitHub

# HƯỚNG DẪN SỬ DỤNG GITHUB


- **Bước 1:**


- Tiếp theo, tại bước 1, bạn điền username, email và password.
- Password của bạn nên có ít nhất một ký tự in thường, một chữ số và gồm ít nhất 7 ký tự.
- Bấm Create an account để sang bước 2


GitHub, Inc.

# Join GitHub

The best way to design, build, and ship software.

 **Step 1:**  
Set up a personal account

 **Step 2:**  
Choose your plan

 **Step 3:**  
Tailor your experience

## Create your personal account

**Username**

lvman ✓

This will be your username — you can enter your organization's username next.

**Email Address**

manleviethuy@gmail.com ✓

You will occasionally receive account related emails. We promise not to share your email with anyone.

**Password**

.....

Use at least one lowercase letter, one numeral, and seven characters.

By clicking on "Create an account" below, you are agreeing to the [Terms of Service](#) and the [Privacy Policy](#).

Create an account

### You'll love GitHub

Unlimited collaborators

Unlimited public repositories

✓ Great communication

✓ Frictionless development

✓ Open source community

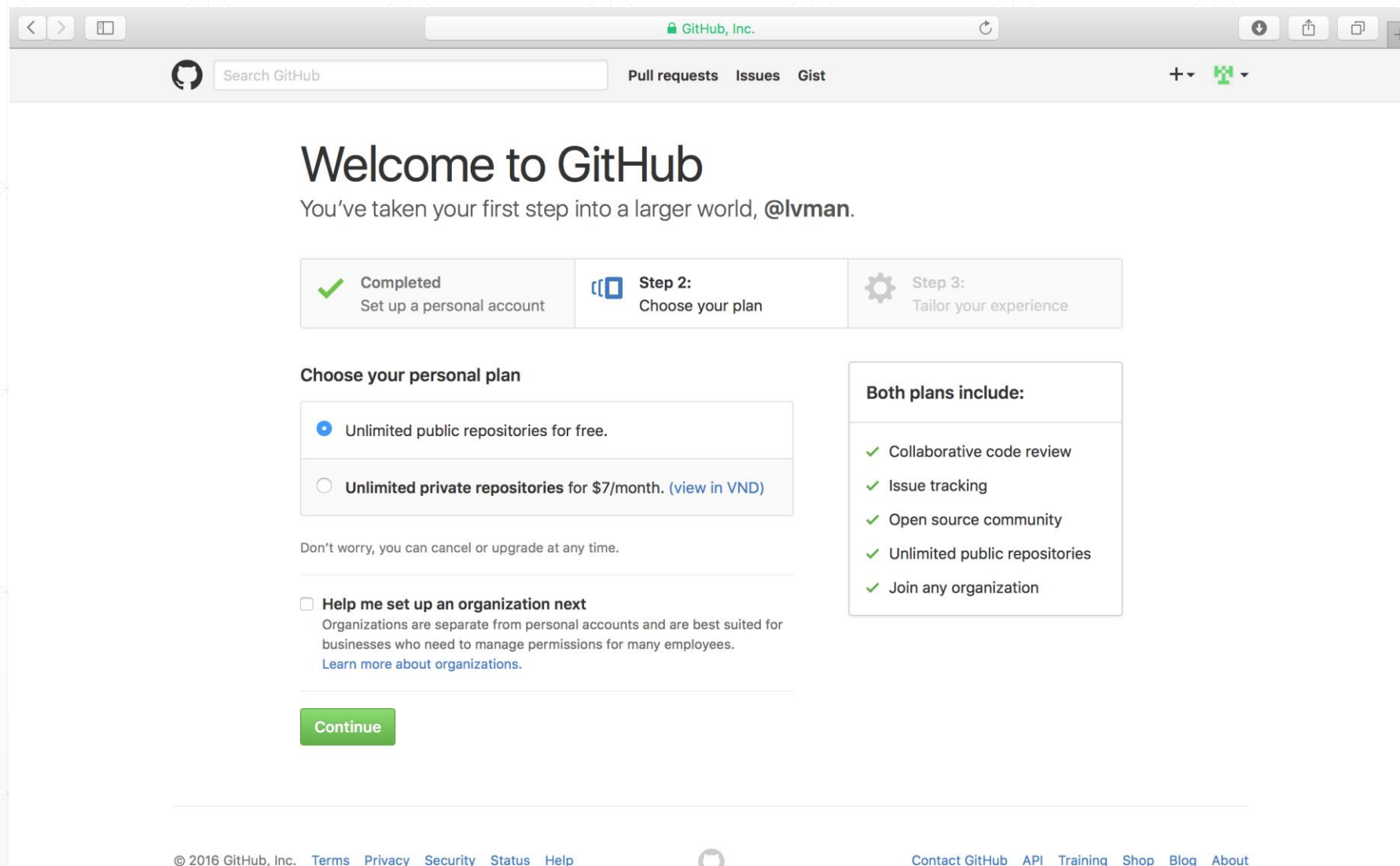
Tài khoản của GitHub



# HƯỚNG DẪN SỬ DỤNG GITHUB

## ▪ Bước 2:

- Tại bước 2, bạn sẽ lựa chọn loại tài khoản.
- Hiện tại, GitHub cung cấp hai loại tài khoản là: miễn phí cho việc lưu trữ các dự án mở (chọn Unlimited public repertoires for free) và có phí (7 đô/tháng) cho lưu trữ các dự án đóng (chọn Unlimited private repertoires for \$7/month).
- Bấm Continue để sang bước 3.



## Tài khoản của GitHub



# HƯỚNG DẪN SỬ DỤNG GITHUB

## ▪ Bước 3:

- Ở bước 3, bạn trả lời một số câu hỏi điều tra của GitHub, rồi bấm Submit.
- GitHub sẽ gửi cho bạn một email vào địa chỉ mail của bạn đã cung cấp cho GitHub để xác thực. Trong email đó, bạn bấm đường link Verify email address để xác thực. Lúc này, GitHub sẽ gửi email chúc mừng và cho phép bạn tạo repertoire (kho) đầu tiên trên GitHub.

<>📄

GitHub, Inc.

⬇️🔗📄+

# Welcome to GitHub

You'll find endless opportunities to learn, code, and create, @lvman.

✓ Completed  
Set up a personal account

📄 Step 2:  
Choose your plan

⚙️ Step 3:  
Tailor your experience

How would you describe your level of programming experience?

☒ Totally new to programming    ☐ Somewhat experienced    ☐ Very experienced

What do you plan to use GitHub for? (check all that apply)

☐ Project Management    ☐ Development    ☐ Research

☒ School projects    ☐ Design    ☐ Other (please specify)

Which is closest to how you would describe yourself?

☐ I'm a hobbyist    ☒ I'm a student    ☐ I'm a professional

☐ Other (please specify)

What are you interested in?

c# ✕

e.g. tutorials, android, ruby, web-development, machine-learning, open-source

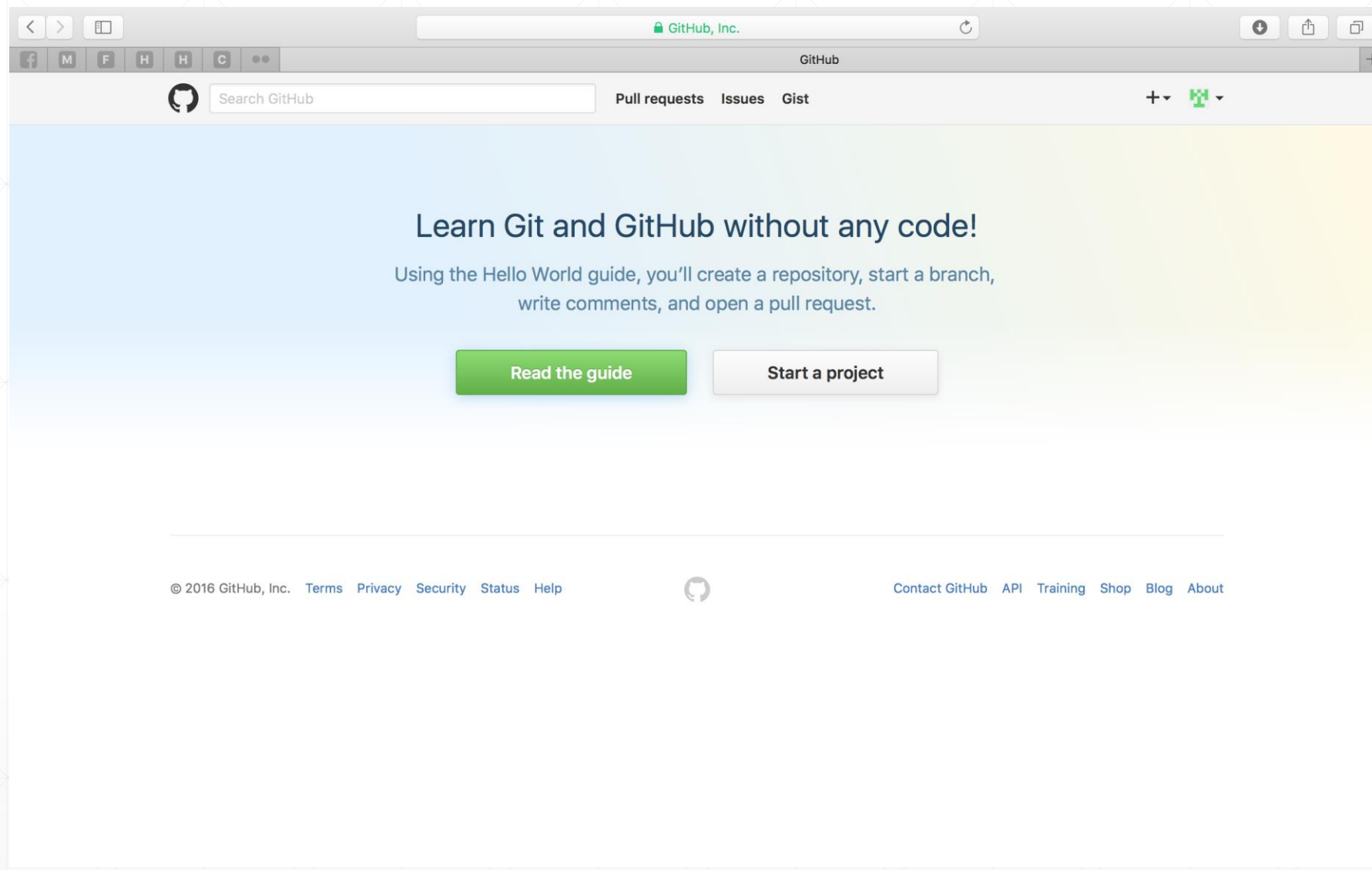
Submit

[skip this step](#)

Tài khoản của GitHub







Chọn Start a project để bắt đầu tạo một repertoire đầu tiên trên GitHub.





# HƯỚNG DẪN SỬ DỤNG GITHUB

- **GitHub Desktop và GitHub Extension for Visual Studio:**

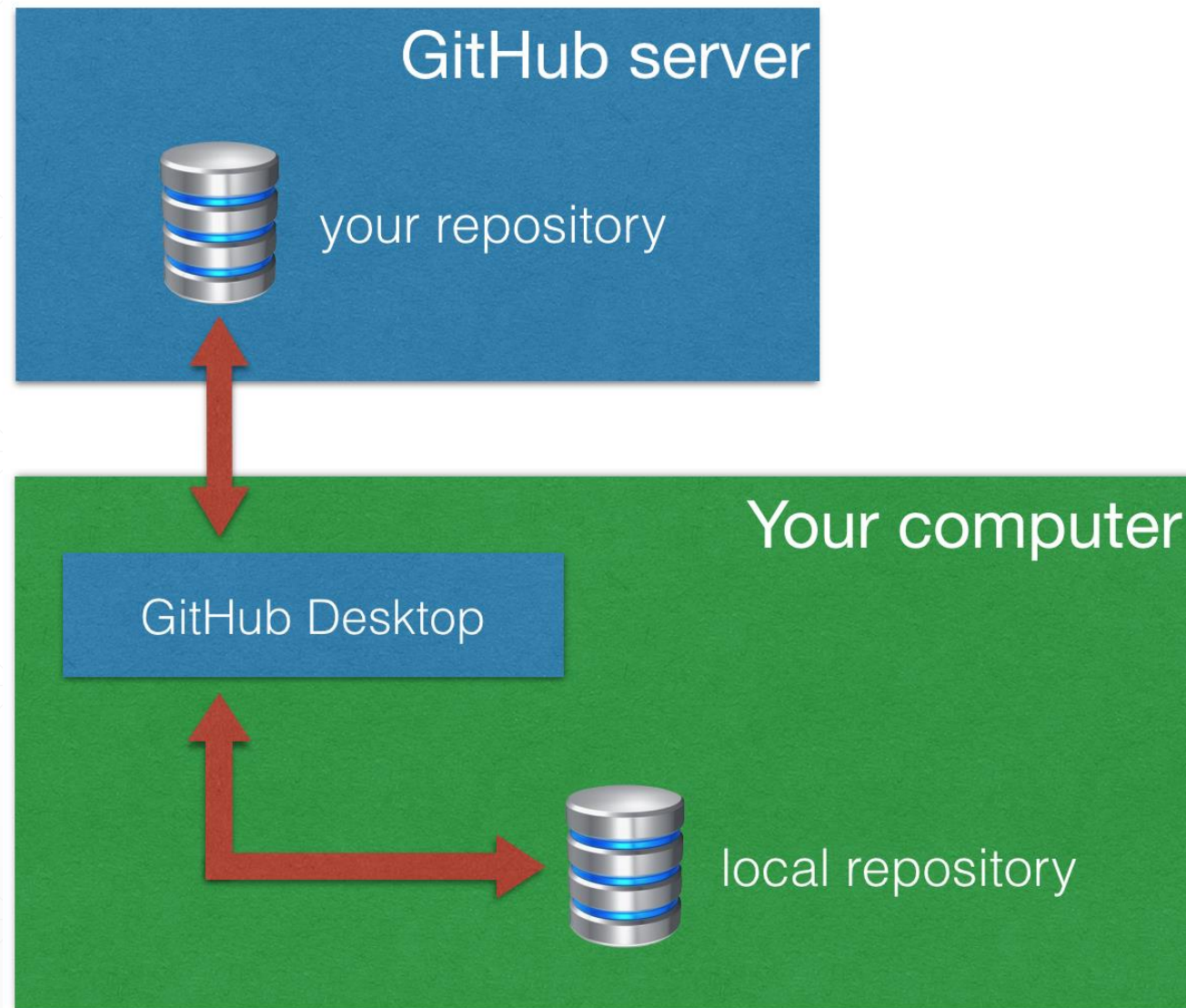
- Bên cạnh việc dùng web để quản lý các dự án, bạn có thể sử dụng phần mềm GitHub Desktop để đồng bộ với các dự án trên GitHub về máy của mình.
- Nếu bạn sử dụng Visual Studio 2015 trở lên, bạn có thể tải về GitHub Extension for Visual Studio. Ứng dụng này tích hợp trực tiếp vào giao diện Visual Studio và cũng hỗ trợ đầy đủ quy trình làm việc, quản lý dự án với GitHub ngay trên giao diện của Visual Studio.

# HƯỚNG DẪN SỬ DỤNG GITHUB

- **GitHub Desktop và GitHub Extension for Visual Studio:**

- Một điểm cần lưu ý khi làm việc với GitHub Desktop và GitHub Extension (cũng như Git) đó là khi bạn sao chép (clone) một kho (repository) với GitHub thì kho đó sẽ được sao chép một bản về trên máy của bạn và bản đó tồn tại độc lập với bản ở trên GitHub.

- Thao tác này khác với tính năng đồng bộ thông thường mà chúng ta hay thấy. Do đó, mọi thao tác sửa đổi, lưu trạng thái (commit) sẽ chỉ được thực hiện ở máy của bạn. Chỉ đến khi bạn đồng bộ (publish hoặc sync) với máy chủ GitHub thì những sửa đổi và lần lưu trạng thái đó mới được chuyển lên GitHub.



GitHub Desktop.

# QUY TRÌNH LÀM VIỆC VỚI GITHUB

1. Tạo một kho (repository) mới hoặc phân tách (fork) một kho đã có;
2. Tạo dự án trong tab Project, chuyển đổi các nhiệm vụ thành các vấn đề (issue);
3. Quản lý các vấn đề (issue), gán nhãn, gán mốc, gán người phụ trách;
4. Lựa chọn vấn đề (issue) cần giải quyết, kéo nhiệm vụ (task) liên quan sang cột In Progress;
5. Sao chép (clone) về máy tính hoặc đồng bộ (sync) lại kho;

# QUY TRÌNH LÀM VIỆC VỚI GITHUB

6. (Nếu cần thiết) Tạo nhánh (branch) mới để giải quyết vấn đề
7. Chỉnh sửa và thực hiện lưu trạng thái (commit)
8. Tạo một yêu cầu gộp (pull request)
9. Kiểm duyệt yêu cầu gộp (review changes): chấp nhận gộp hoặc không chấp nhận gộp, đóng yêu cầu gộp
10. Trộn (merge) yêu cầu gộp vào nhánh chính
11. Xoá nhánh (branch) nếu muốn
12. Đóng vấn đề (issue) liên quan, kéo nhiệm vụ (task) liên quan sang cột Done
13. Quay lại bước 4.