**Bài tập thực hành 06**

Làm quen với Github

**Mục tiêu cần đạt**:

Sau buổi các bạn cần nắm thao tác sử dụng github

Các bạn đọc và thực hiện viết code như hướng dẫn để làm quen, sau đó đọc trả lời các **Yêu cầu** ở dưới bài hướng dẫn để thực hiện yêu cầu bài thực hành.

**Gợi ý Tham khảo:**

* <https://docs.github.com/en/get-started/quickstart/hello-world>
* https://www.youtube.com/watch?v=8Dd7KRpKeaE
* https://topdev.vn/blog/github-la-gi/
* và các nguồn khác mà sinh viên tìm được
* Các lệnh trong git https://git-scm.com/docs

## Github

Là nơi các bạn có thể dùng làm nơi lưu trữ code, csdl để làm việc theo nhóm hiệu quả. Để thực hiện tạo các kho repository để lưu trữ dữ liệu bạn sẽ cần tạo 1 tài khoản github tại <http://github.com>.

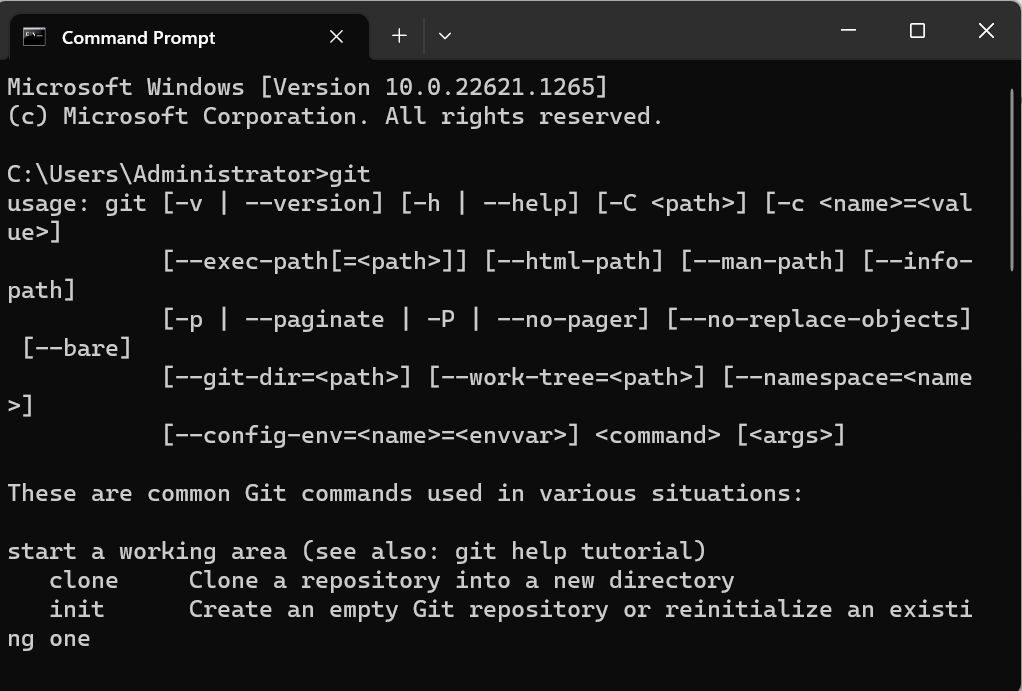
Bạn hãy đọc yêu cầu bài thực hành và theo đó bạn sẽ tìm hiểu kỹ về việc dùng github. Nếu chưa có git bạn download tại đây: <https://git-scm.com/download/win> và cài đặt.

Sau đó vào cmd gõ “git để kiểm tra việc cài đặt, nếu hiện ra được thông báo này là đã cài đặt thành công:



# Yêu cầu bài thực hành:

1. Bạn theo link sau: <https://www.techrepublic.com/article/how-to-install-github-desktop/>, theo hướng dẫn trong link để tiến hành cài đặt git trên máy.



1. Bạn hãy tìm hiểu và liệt kê những lợi ích của git và github.

* GIT là hệ thống quản lý các phiên bản dưới dạng phân tán. Đây là hệ thống quản lý phổ biến nhất hiện nay. GIT là một phiên bản của Distributed Version Control System – DVCS hay còn gọi là VCS. Sử dụng GIT sẽ đem tới cho các lập trình viên một kho lưu trữ, trong đó chứa đầy đủ các lịch sử thay đổi của hệ thống.
* Lợi ích của GIT:
  + Thao tác thực hiện nhanh, gọn, lẹ hơn, việc sử dụng an toàn hơn.
  + Người dùng có thể dễ dàng kết hợp các nhánh với nhau. Qua đó giúp đơn giản hóa quy trình code theo nhóm.
  + Khi sử dụng GIT, người dùng chỉ cần sử dụng clone mã nguồn từ khoa chứa/phiên bản nào đó có sẵn trong kho lưu trữ/ một nhánh trong kho lưu trữ. Hệ thống lưu trữ chính xác sẽ cho phép bạn có thể làm việc bất cứ lúc nào, ở bất cứ đâu.
  + Giờ đây, việc lập trình các dự án song song sẽ cực kỳ dễ dàng, hiệu quả và chính xác.
* GitHub giống như một trang mạng xã hội dành riêng cho lập trình viên, là hệ thống quản lý dự án và phiên bản code.
* GitHub còn là dịch vụ cung cấp kho lưu trữ mã nguồn Git cho các phần mềm.
* Ngoài các tính năng cơ bản của Git, GitHub còn được hỗ trợ thêm các tính năng về social networking như followers, feeds, network graph, giúp cho các developer dễ dàng tương tác cùng nhau.
* Lợi ích của Github:

### Quản lý source code dễ dàng

### Tracking sự thay đổi qua các version

### Tối ưu hiển thị tài liệu bằng Markdown

### Github giúp cải thiện kỹ năng code, thậm chí là tracking bug

### Github là một kho tài nguyên tuyệt vời

### Github Action

### Github Package Registry

### Mở rộng mối quan hệ, kết nối với người cùng ngành

* + Làm đẹp CV xin việc cho các lập trình viên
  + Có thêm một kho tài nguyên chất lượng

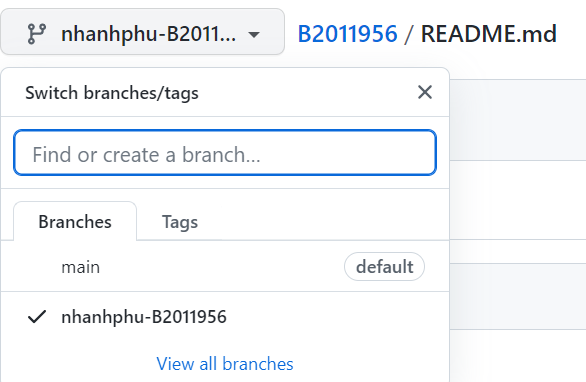
1. Repositories trong github là gì (Tham khảo [[1]](#footnote-1))? Có thể chứa được tối đa dung lượng bao nhiêu miễn phí? Cho biết kích thước tập tin lớn có thể tải lên repository github. Tham khảo ở[[2]](#footnote-2), bạn hãy tạo 1 repository với tên là mã số sinh viên của bạn. Trong file readme bạn hãy giới thiệu thông tin về bạn. Chụp hình trang readme đã tạo và dán đường link vào bài tập này

* Repository được hiểu là một kho lưu trữ nơi chứa các files của dự án. Các file đó có thể là code, hình ảnh, âm thanh hoặc mọi thứ liên quan đến dự án. Bạn có thể tổ chức kho lưu trữ của mình dưới nhiều hình thức khác nhau, hai loại kho lưu trữ trong Github là Local Repository và Remote Repository.
* Dung lượng tối đa có thể chứa:
  + Các tệp văn bản trên 512 KB luôn được hiển thị dưới dạng văn bản thuần túy.
  + Các tệp văn bản trên 5 MB chỉ có sẵn thông qua các URL thô của chúng, được cung cấp qua raw.githubusercontent.com; ví dụ, <https://raw.githubusercontent.com/octocat/Spoon-Knife/master/index.html>.
  + Trong yêu cầu kéo, tổng số khác biệt không được vướt qua 20.000 dòng mà bạn có thể tải hoặc 1MB dữ liệu khác biệt.
  + Không có điểm khác biệt nào của một tệp có thể vượt quá 200.000 dòng mà bạn có thể tải hoặc 500 KB dữ liệu khác biệt thô. Bốn trăm dòng và 20 KB được tải tự động cho một tệp.
  + Số lượng tệp tối đa trong một khác biệt du nhất được giới hạn ở 300
  + Số lượng tệp có thể kết xuất tối đa (chẳng hạn như hình ảnh, tệp PDF và tệp GeoJSON) trong một điểm khác biệt duy nhất được giới hạn ở 25 tệp.
  + Github có hỗ trợ Git LFS, nhưng chỉ cho phép 10 GB trên mỗi kho lưu trữ. Điều này áp dụng cho cả repo thông thường và repo LFS. Tuy nhiên, khá dễ dàng để mua thêm dữ liệu từ Cài đặt> Thanh toán và 50GB chỉ phải trả thêm 4,20 đô la mỗi tháng.
  + Readme: *https://github.com/qanh10/B2011956/blob/main/README.md*



1. Branch trong git là gì? Bạn hãy tạo 1 branch tên “nhanhphu-<mssv của bạn>”

* Branch: được hiểu là các nhánh. Các Branch sẽ tương ứng với các phiên bản cụ thể trong kho lưu trữ. Branch cho phép người dùng có thể dễ dàng truy cập, theo dõi các thay đổi của các phiên bản, từ đó thử nghiệm hoặc lựa chọn sử dụng phiên bản cũ hơn.



1. Hãy cho biết hành động commit, push, pull trong git là gì?

* Commit: Commit là một thuật ngữ cực kỳ quen thuộc trong Git. Vậy commit trong git là gì? Hiểu một cách đơn giản, commit là thuật ngữ thể hiện một thời điểm cụ thể trong lịch sử thực hiện code của bạn. Các dự án sẽ có nhiều thời điểm cụ thể trong lịch sử nên sẽ có nhiều code. Khi sử dụng git, bạn có thể dùng lệnh commit kết hợp với lệnh git add để kiểm tra các thay đổi và lưu trữ trong local repository.
* Push: Lệnh Push được sử dụng khi người lập trình cần cập nhật các branch từ xa. Đồng thời, Git Push cũng được dùng cho những thay đổi mới nhất mà người sử dụng đã commit.
* Pull:
  + Pull là việc người dùng đề xuất các thay đổi mới cho Master Branch. Đây là tính năng phù hợp với các dự án cần làm việc nhóm. Người thực hiện có thể dùng tính năng Pull Request để yêu cầu người có nhiệm vụ thực hiện bảo trì kho lưu trữ để xem xét các thay đổi của hệ thống.
  + Người dùng có thể dùng lệnh Git Pull để thay đổi lịch sử lưu trữ vào các nhánh chính.

1. Hãy thực hiện hành động upload tập tin thực hành buổi 5 lần trước lên github và commit.
2. Thực hiện lệnh git clone … để download code từ repository bạn tạo ở câu số 3.



1. Chép 1 tập tin thực hành buổi 4 vào thư mục vừa download xuống và thực hiện lệnh

git add .

git commit -m “upload du lieu”

git push

Hãy cho biết ý nghĩa 3 lệnh trên. Và quan sát kết quả trên trang github của bạn đã tạo ở câu 3 và nhận xét.

* Git Add:
  + Được sử dụng khi cần đổi Stage hoặc index đến thư mục bạn đang làm việc.
  + Thêm một lệnh Git add. Tuy nhiên, lệnh git add này sẽ được sử dụng để đưa tệp tin vào khu vực staging. Khi tìm hiểu git là gì, bạn cần lưu ý sự khác biệt giữa 2 lệnh git add để sử dụng chính xác.
  + Để thêm tệp tin, bạn có thể chọn git add tên file hoặc all nếu muốn thêm tất cả các file có trong thư mục.
* Git push: Lệnh Push hay pull cho phép người sử dụng có thể thay đổi tới các kho lưu trữ remote. Nếu bạn đã thêm các thay đổi và muốn đưa các thay đổi đó lên đầu, bạn có thể dùng lệnh này để cập nhật và apply các thay đổi đó thông qua việc code. Bạn có thể dùng bằng code: *git pull <:remote:> <:branch:> and git push <:remote:> <:branch:>.*

1. Upload tập tin thực hành buổi 3 lên trang github của bạn và thực hiện lệnh

git pull

Quan sát kết quả đạt được và nhận xét.

1. Bạn hãy add các contributor là những bạn sẽ làm cùng nhóm với bạn



1. Tham khảo ở: <https://www.makeareadme.com/>, hãy trang trí trang readme của bạn sao cho đẹp với các mô tả thông tin bài tập nhóm.

Chú ý:

* Các bạn nộp file word: Đặt tên file: <mssv><hoten><bai5>.docx nộp lên classroom, kèm với các file code đã tạo (**ngoại trừ file word trả lời câu hỏi, các file còn lại các bạn nén vào file .zip**).
* Mỗi câu các bạn trả lời bằng hình/trả lời dạng text tùy vào yêu cầu của câu hỏi, và ghi rõ số thứ tự câu hỏi mà bạn trả lời.
* Các bài phát hiện copy từ các bài của bạn khác sẽ không được tính điểm.
* *Chú ý: Các câu trả lời có tham khảo trên Internet phải trích dẫn link*

1. https://docs.github.com/en/repositories/creating-and-managing-repositories/about-repositories [↑](#footnote-ref-1)
2. https://docs.github.com/en/get-started/quickstart/hello-world [↑](#footnote-ref-2)