**2. Git & HTML**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Khái niệm về Git? Phân biệt với Github | Git: là công cụ quản lý mã nguồn theo hình thức phân tán.  Có hai loại respository của git đó là :   * Local Repository : ở trên máy của lập trình viên * Remote Repository : ở trên máy chủ chia sẻ   Lợi ích sử dụng :   * Giảm conflic khi thực hiện dự án theo nhóm * Lưu trữ mã nguồn an toàn * Có thể phục hồi các phiên bản khác nhau * Có thể kiểm tra ai là người gây ra lỗi khi thực hiện dự án * Có thể khôi phục lại các file lỡ xóa mất   Phân biệt Git và Github :   |  |  | | --- | --- | | Git | Github | | Là công cụ quản lý mã nguồn phân tán | Là dịch vụ lưu trữ mã nguồn trực tuyến dựa trên Git | | Theo dõi lưu trữ và quản lý lịch sử các phiên bản của mã nguồn dự án | Cung cấp một nền tảng để lưu trữ mã nguồn, theo dõi sự thay đổi, và cộng tác với người khác trên dự án mã nguồn mở hoặc dự án riêng tư. | |
| Các câu lệnh cơ bản của Git | 1. Khởi tạo một kho lưu trữ (repository) Git:    * **git init** 2. Sao chép một kho lưu trữ từ xa vào máy tính:    * **git clone <URL\_repository>** 3. Kiểm tra trạng thái làm việc:    * **git status** 4. Thêm các tệp vào vùng lưu trữ (staging area) để chuẩn bị commit:    * **git add <tên\_tệp>** 5. Commit lưu các thay đổi đã thêm vào vùng lưu trữ với một thông điệp:    * **git commit -m "Thông điệp commit"** 6. Đẩy các thay đổi lên kho lưu trữ từ xa (remote repository):    * **git push** 7. Cập nhật kho lưu trữ từ xa với phiên bản mới nhất:    * **git pull**   Ngoài ra còn có một số câu lệnh khác, xem thêm trên Internet. |
| HTML | HTML là viết tắt của "Hypertext Markup Language" (Ngôn ngữ Đánh dấu Siêu văn bản). Đó là một ngôn ngữ đánh dấu được sử dụng để tạo và cấu trúc các trang web trên internet. |
| Các thẻ HTML thông dụng | 1. **<html>**: Thẻ gốc của tài liệu HTML, đánh dấu đầu và cuối của một trang web. 2. **<head>**: Khu vực đặc biệt chứa các thông tin về trang web, chẳng hạn như tiêu đề và các liên kết đến tệp CSS hoặc JavaScript. 3. **<title>**: Xác định tiêu đề của trang web, hiển thị trên thanh tiêu đề của trình duyệt. 4. **<meta>**: Được sử dụng để định nghĩa thông tin về trang web, chẳng hạn như mã ngôn ngữ và mã ký tự. 5. **<link>**: Dùng để liên kết tài liệu với các tệp CSS hoặc biểu đồ khác để tùy chỉnh giao diện trang web. 6. **<script>**: Sử dụng để nhúng mã JavaScript vào trang web để thêm các chức năng tương tác. 7. **<body>**: Chứa nội dung chính của trang web, chẳng hạn như văn bản, hình ảnh, liên kết và nhiều phần tử khác. 8. **<h1>**, **<h2>**,..., **<h6>**: Được sử dụng để tạo tiêu đề với mức độ ưu tiên khác nhau, **<h1>** là tiêu đề quan trọng nhất và **<h6>** là tiêu đề ít quan trọng hơn. 9. **<p>**: Được sử dụng để định dạng đoạn văn bản. 10. **<a>**: Tạo liên kết đến trang web khác hoặc tài liệu. 11. **<img>**: Hiển thị hình ảnh trên trang web. 12. **<ul>** và **<ol>**: Sử dụng để tạo danh sách không có thứ tự (unordered list) và danh sách có thứ tự (ordered list) tương ứng. 13. **<li>**: Sử dụng để định dạng các mục danh sách trong **<ul>** hoặc **<ol>**. 14. **<div>**: Được sử dụng để tạo các phần tử chứa hoặc nhóm các phần tử HTML khác để tạo cấu trúc và định dạng trang web. 15. **<span>**: Được sử dụng để đánh dấu một phần của văn bản trong một đoạn văn bản dài.   Ngoài ra vẫn còn nhiều thẻ khác, xem thêm trên Internet |
| Entity trong HTML | Entity là các từ được quy định sẵn trong HTML dùng để hiển thị một số ký tự đặc biệt. Các entity bắt đầu bằng dấu “&” và kết thúc bằng dấu “;” |

# **HƯỚNG DẪN MẪU:**

[Thực hành] Tạo dự án mới trên GitHub

Mục tiêu

Luyện tập thao tác tạo dự án mới trên GitHub.

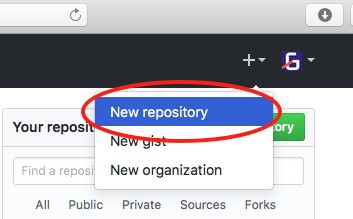
# Mô tả

Trong phần này, chúng ta sẽ luyện tập thao tác quan trọng khi làm việc với GitHub đó là tạo một dự án mới.

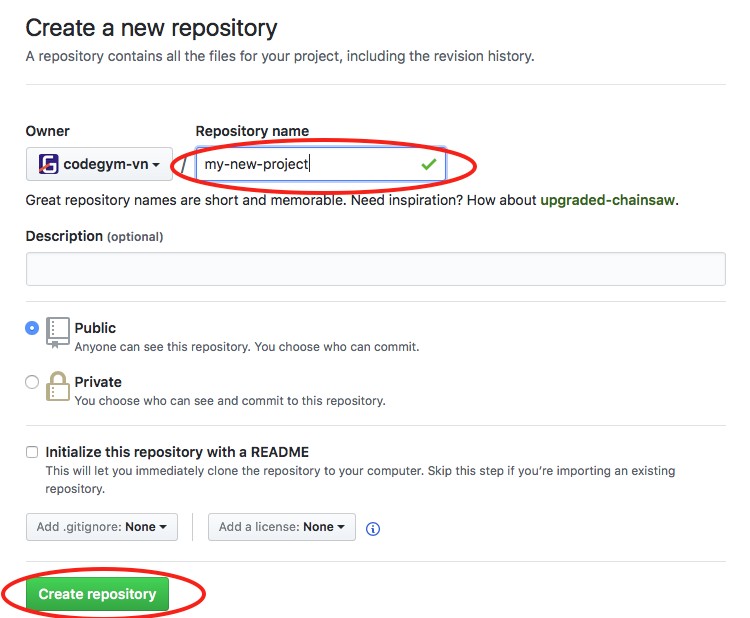
# Hướng dẫn

Trong phần hướng dẫn này, chúng ta sẽ tạo một dự án mới trên GitHub và bắt đầu đưa mã nguồn lên đó. Các bước thực hiện như sau:   
1. Đi đến trang: *https://*[*github.com*](https://github.com/)

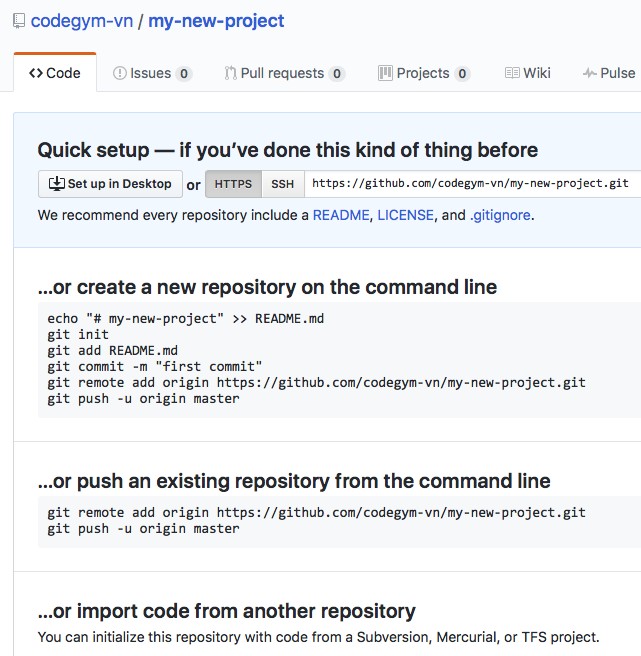
1. Nhấn vào mục "New repository" để bắt đầu tạo một Repository mới:



Trong màn hình tạo Repository mới, chúng ta điền tên của dự án mới vào. Chẳng hạn chúng ta đặt tên dự án mới là "*my-new-project*". Sau đó nhấn và nút "Create repository".



Kết quả sẽ được một Repository mới. Giao diện ban đầu của Repository này chứa các hướng dẫn cần thiết để làm việc với nó:



1. Sau khi đã tạo được Remote Repository trên GitHub, bây giờ chúng ta sẽ mở cửa sổ Terminal lên để tạo Local Repository và đồng bộ nó tới Remote Repository trên.

Chúng ta sẽ tạo một thư mục mới trên máy tính của mình, có thể đặt tên bất kỳ. Chẳng hạn chúng ta có thể đặt tên thưc mục là "*my-new-*

*project*" (giống với tên của Repository). 

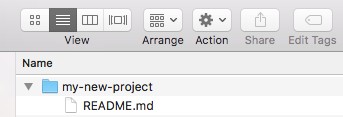
1. Chuyển Terminal đến thư mục "*my-new-project*" vừa tạo và sử dụng lệnh *git init* để khởi tạo Repository mới.
2. Tiếp theo, sử dụng lệnh *git remote add origin* để kết nối tới Remote Repository (chú ý sử dụng dúng đường dẫn của Repository mà bạn vừa tạo):

git remote add origin https://github.com/codegym-vn/my-new-project.git



Như vậy là Local Repository và Remote Repository đã được liên kết với nhau. Bây giờ chúng ta có thể bắt đầu làm việc trên dự án của mình.

1. Trong thư mục "*my-new-project*", chúng ta tạo một le mới có tên là "README.md". Đây là le dùng để chứa mô tả về dự án này, hầu hết tất cả các Repository đều có le này để giới thiệu cho người khác biết về mục đích của dự án. Chúng ta có thể viết một nội dung bất kỳ vào le này.



1. Sử dụng câu lệnh *git add* để đưa le README.md vào trong chỉ mục của Git.

git add README.md



1. Sử dụng câu lệnh *git commit* để lưu lại các thay đổi.

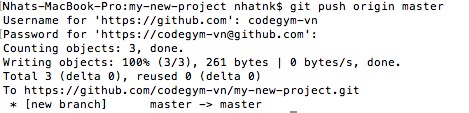
git commit -m "Add README.md file"



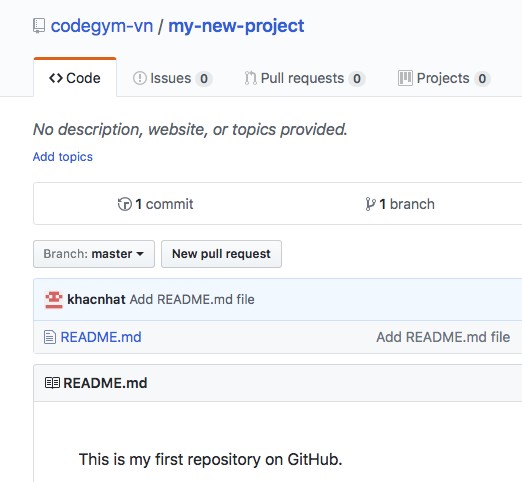
Trong câu lệnh trên, tuỳ chọn *-m* được sử dụng để ghi một "*message*" đánh dấu cho thay đổi vừa rồi. Nó giải thích rằng chúng ta vừa thêm một le README.md mới. Chúng ta cần ghi các thông điệp có ý nghĩa để dễ dàng cho việc hiểu và cộng tác nhóm sau này.

1. Sử dụng câu lệnh *git push* để đẩy các thay đổi ở Local Repository lên trên Remote Repository.

git push origin master



Như vậy là chúng ta đã hoàn thành một luồng làm việc cơ bản với GitHub. Local Repository và Remote Repository đã được đồng bộ với nhau.

Thử quay lại trang dự án ở trên GitHub thì chúng ta sẽ thấy một giao diện như sau:  


11. Hãy lặp lại các bước từ 7 đến 10 để tạo các file mới.

[Thực hành] Clone dự án từ Git

Mục tiêu

Luyện tập thao tác Clone dự án có sẵn từ GitHub.

# Mô tả

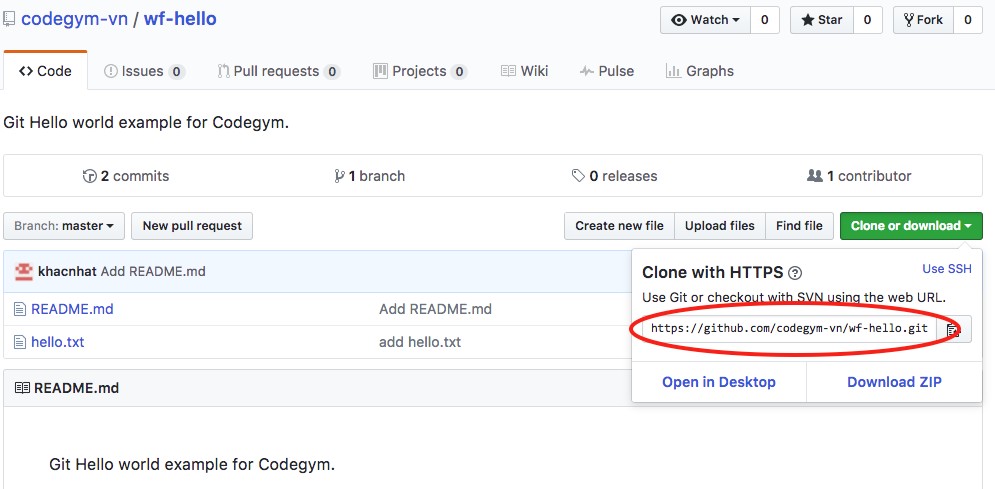
Trong phần này, chúng ta sẽ luyện tập thao tác cơ bản nhất khi làm việc với Git đó là sao chép một Repository có sẵn về máy của mình.

Sau khi thực hiện xong các bước hướng dẫn thì hãy hoàn thành bài thực hành này bằng cách mô tả lại cú pháp và ý nghĩa của câu lệnh clone.

# Hướng dẫn

Trong phần hướng dẫn này, chúng ta sẽ sao chép (clone) một Repository có sẵn từ GitHub về máy của mình. Các bước thực hiện như sau:

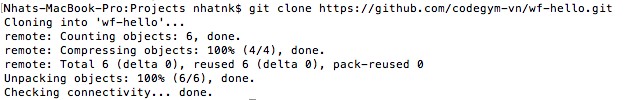
1. Đi đến trang [*https://github.com/codegym-vn/wf-hello*](https://github.com/codegym-vn/wf-hello)
2. Nhấn vào nút "Clone or download" rồi copy đường dẫn của Repository



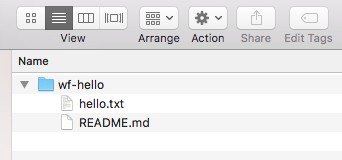
1. Mở màn hình Terminal trên máy tính của mình, di chuyển đến thư mục mình muốn chứa các Repository (sử dụng lệnh *cd*).
2. Dùng lệnh *git clone* để sao chép Repository về máy của mình:

git clone https:*//github.com/codegym-vn/wf-hello.git*

Kết quả:



1. Mở thư mục vừa sao chép về, chúng ta sẽ thấy các file nằm trong đó:



Như vậy, trong phần này chúng ta đã học cách để sử dụng câu lệnh *git clone* để sao chép một Repository có sẵn.