**Git Bangla tutorial: গিট এবং গিটহাব: নতুনদের জন্য গিট**

**Git Bangla Tutorial/Github Bangla Tutorial/ Git and GitHub/What is Git**

**Git Bangla tutorial:** ভার্সন কন্ট্রোল (**Version control**) এর জন্য গিট একটি জরুরি সফটওয়ার। গিট এবং গিটহাব (**Git and GitHub**) বড় আকারে টিমওয়ার্ক করার জন্য খুবই জরুরি সংযোজন। একইসাথে অসংখ্য ওপেন-সোর্স প্রজেক্টে কাজ করার জন্য একটি মাধ্যম গিট এবং গিটহাব। তাই বর্তমানে টিমওয়ার্ক করার জন্য গিট এবং গিটহাবে নিজেকে অভ্যস্ত করা খুবই জরুরি। তাই গিট এবং গিটহাব নিয়ে একটি সংক্ষিপ্ত আকারে লিখার চেষ্টা করেছি।

**Git Bangla tutorial**

আজকে আমরা গিট এবং গিটহাবের বেসিক (**Git and GitHub basics**) নিয়ে জানবো, কিভাবে আপনার লোকাল কম্পিউটারে একটি প্রজেক্ট বানাবেন এবং তাকে গিটহাবে রাখবেন। আরও দেখবো কিভাবে আমরা আমাদের লোকাল প্রজেক্টের সাথে আমরা গিটহাবকে সিনক্রোনাইজ করবো। তাহলে শুরু করা যাক।

**ভার্সন কন্ট্রোল কি? (What is version control)**

**ভার্সন কন্ট্রোল কি? আমাদের কেন ভার্সন কন্ট্রোল দরকার? (What is version control and why we should care?)**>> যে পদ্ধতিতে কোনও প্রজেক্টের ফাইলগুলোর পরিবর্তন ট্র্যাক করে রাখা হয় যাতে করে আমরা পরে দরকারে আগের নির্দিস্ট ভার্সনে যেতে পারি তাকে ভার্সন কন্ট্রোল (**Version Control System**বা **VCS**) বলে। **গিট প্রায় যেকোনো ফাইলের পরিবর্তন গুলো ট্র্যাক করে রাখতে পারে।**

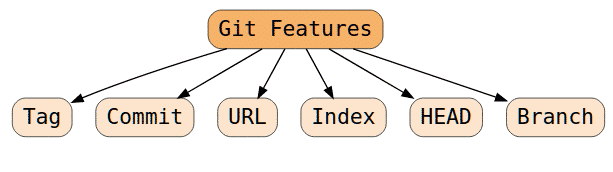
আপনি যদি কোন ওয়েব ডিজাইনার বা গ্রাফিক ডিজাইনার হন তাহলে আপনি চাইবেন আপনার প্রতিবারের করা ডিজাইন (ডিজাইন এর প্রতিটি ভার্সন) যেন সংরক্ষিত থাকে। এই কাজের জন্যই একটি ভার্সন কন্ট্রোল সিস্টেম (VCS) ব্যবহার করা হয়। যদি কখনো আপনার নতুন আপডেটে কোন সমস্যা হয় তাহলে আপনি version control system বা VCS ব্যবহার করে আমরা সিলেক্টেড ফাইলকে আগের অবস্থায় নিয়ে যেতে বা সম্পুর্ন প্রজেক্টকে আগের ভার্সনে নিয়ে যেতে পারি বা রিভার্ট (Revert) করতে পারি।

একই সাথে আমরা ভার্সন কন্ট্রোল সিস্টেম ব্যবহার করে আমাদের প্রজেক্টের ফাইল গুলো আগের ভার্সনের সাথে তুলনা করতে পারি। আমাদের প্রজেক্টে নতুন কিছু যুক্ত করার পরে যদি কোন সমস্যা তৈরি হয় বা ভুলে কিছু ডিলিট হয়ে যায় তাহলে আমরা **Version Control System** ব্যবহার করে **খুব সহজেই আগের অবস্থায় ফিরে যেতে পারি।**

**গিট কি? (What is Git?)**

গিট হলো একটি Open-source ভার্সন কন্ট্রোল সিস্টেম যা দিয়ে আমরা একটি প্রজেক্টের ফাইল গুলোর পরিবর্তন ট্র্যাক করতে পারি। গিট ২০০৫ সালে লিনাক্স কার্নেলের জন্য প্রকাশ করা হয় যার স্রস্টা Linus Torvalds. অন্য কার্নেল ডেভেলপাররা এটার ডেভেলপমেন্ট এ কন্ট্রিবিউট করে। বর্তমানে এটা বিশ্বের সবচেয়ে জনপ্রিয় ভার্সন কন্ট্রোল সিস্টেম। git আপনার প্রজেক্টের লাস্ট স্টেট (snapshot) সেভ করার মাধ্যমে কাজ করে। লাস্ট snapshot থেকে যেসব ফাইল পরিবর্তন হয়েছে বা তৈরি হয়েছে গিট সেসব ফাইলকে রেখে দেয়।

**গিটের পরিভাষা পরিচিতি (Git terminology)**



**Git Bangla tutorial: Git states কি?**

গিটের তিনটি প্রাইমারি স্টেট রয়েছে। নিছে তিনটি স্টেট বর্ননা করা হলো।

1. মডিফাইড – Modified **:** ফাইল পরিবর্তন করা হয়েছে, কিন্তু গিট এখন এইগুলো মার্ক করেনি। এই ফাইলগুলো গিটের পরবর্তি স্নাপশটে অন্তর্ভুক্ত হবেনা।
2. স্টেজড – Staged: ফাইলের পরিবর্তন গুলো গিট ট্র্যাক করে রেখেছে, এবং এদের পরবর্তি গিট স্নাপশটে অন্তর্ভুক্ত করা যাবে।
3. কমিটেড – Committed: ফাইল গুলো পরবর্তি গিট স্নাপশটের অন্তর্ভুক্ত করা হয়েছে।

***স্নাপশট বলতে কি বুঝাচ্ছি?***

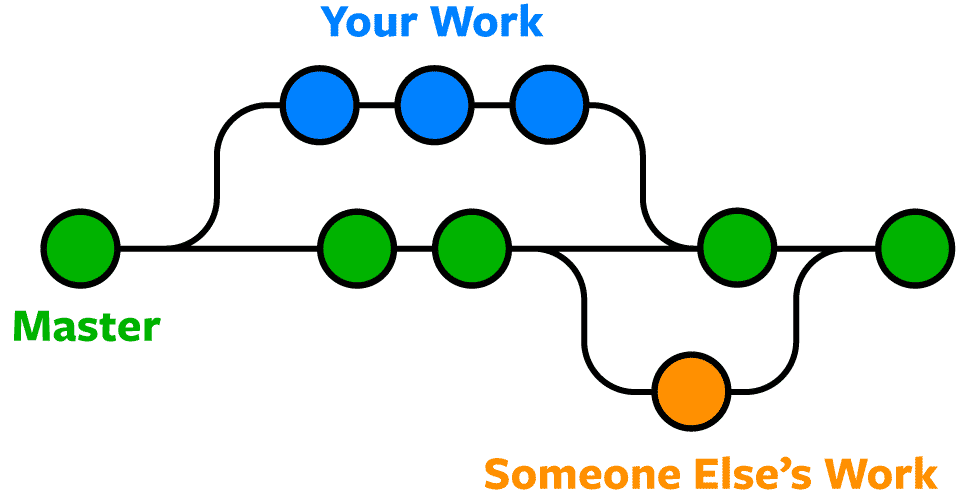
*একটি গিট স্নাপশট কোন নির্দিস্ট সময়ের সমস্ত ফাইল এবং ফোল্ডার ধারণ করে। কনেটেন্ট গুলোকে****গিট ট্রি অবজেক্ট****দ্বারা উপস্থাপন করা হয় এবং****.git****নামে একটি ফোল্ডারে রাখা হয়।*

Stackoverflow.com থেকে

**গিট রিপজিটরি কি? (What is git repository?)**

সহজ ভাষায় বলতে গেলে গিট রিপজিটরি হলো একটি ফোল্ডার বা ডিরক্টরি যেখানে প্রজেক্ট রাখা হয়। এটা আপনার লোকাল কম্পিউটারও হতে পারে বা কোন অনলাইন হোস্টিংও হতে পারে। এই রিপোজিটরি আপনার সম্পুর্ন কোডবেজ এবং প্রত্যেকটি রিভিশন হিস্টরি সেভ করে রাখবে। গিট রিপোজিটরি রাখার জন্য জনপ্রিয় অনলাইন হস্টিং সার্ভিসগুলো হলো, [GitHub](https://github.com/), [GitLab](https://about.gitlab.com/), এবং [Bitbucket](https://bitbucket.org/)

**গিট ব্রাঞ্চ কি? (What is git branch)**

Git Branch

ধরা যাক আমরা আমাদের প্রজেক্টে কোন নতুন ফিচার যুক্ত করতে চাই। এর জন্য আমাদের মূল প্রজেক্টে সরাসরি কোড আপডেট না করে একটা নতুন ওয়ার্কিং ডিরেক্টরি করা হয়। একেই আমরা ব্রাঞ্চ বলি। এখানে আমরা প্রজেক্টে নতুন ফিচার যুক্ত করে টেস্টিং করি এবং সব কিছু ঠিক থাকলে মূল প্রজেক্টে মার্জ করে নিই। গিট ব্রাঞ্চ গুলো স্বাধীনভাবে প্রজেক্টে কাজ করার সুবিধা দেয়। আমরা যদি কোন প্রজেক্টে ব্রাঞ্চ তৈরি করি তবে আমরা নতুন যেসব কাজ করবো তার হিস্টরি/ রেকর্ড এই ব্রাঞ্চেরই অন্তর্ভুক্ত হবে। আমরা git branchকমান্ড ব্যবহার করে নতুন ব্রাঞ্চ তৈরি/নামকরন/ডিলিট করতে পারি।

**গিট কমিট কি? (What is git commit?)**

আমরা গিট কমান্ড ব্যবহার করে আমরা যেকোনো মোমেন্টে আমাদের প্রজেক্টের স্নাপশট রেখে দিতে পারি। এই কমান্ড ই **গিট কমিট কমান্ড**। git commitকমান্ড ব্যবহার করার আগে আমাদেরকে সমস্ত ফাইলকে স্টেজিং এরিয়াতে নিতে হবে। এর জন্য আমাদের git add ব্যবহার করতে হবে। এতে করে আমাদের Modified Files/Folder গুলো কমিটের উপযুক্ত হবে (Staged হবে)। এসব কমান্ড নিয়ে আমরা বিস্তারিত দেখবো একটু পরে।

**গিট চেকআউট কি? (What is git checkout?)**

গিট চেকআউট কমান্ড গিট ব্রাঞ্চ গলোর মধ্যে সুইচ করা এবং ফাইল restore করার জন্য ব্যাবহার করা হয়। সহজ কথায় বলতে গেলে এটাকে প্রজেক্টের বিভিন্ন ভার্সনের মধ্যে সুইচ করার একটি রাস্তা হিসেবে বিবেচনা করা যেতে পারে। এই কমান্ড তিনটি উপদানের উপর অপারেশন করতে পারে। যথা, **ফাইল, কমিট, ব্রাঞ্চ**

**গিট ক্লোন এবং গিট পুল কমান্ড (git clone and git pull)**

দুটি কমান্ডই রিমোট রিপোজিটরি থেকে ফাইল লোকাল কম্পিউটারে ডাউনলোড করতে ব্যাবহার করা হয়। git clone কমান্ড কোন রিমোট রিপোজিটরি থেকে বর্তমান ভার্সন গুলো ডাউনলোড করে লোকাল মেশিনে সেভ করে রাখে। এরপর আপনি এডিট শুরু করতে পারবেন।

অপরদিকে git pull কমান্ড ব্যাবহার করে আমরা রিমোট রিপোজিটরি থেকে ফাইলগুলো ডাউনলোড করে বর্তমান লোকালে সেভ করা ফাইলগুলোকে আপডেট করতে পারি। এটা অনেকটা আমার কাজের সাথে আসল প্রজেক্টের নতুন কাজগুলো মার্জ করার একটি পদ্ধতি।

**গিট পুশ কমান্ড (Git push command)**

গিট পুশ কমান্ডের মাধ্যমে কোন লোকাল প্রজেক্টে অনলাইনে কোন রিপজিটরিতে সেভ করা হয়।

**গিট ট্যাগ কি? (What is git tag?)**

এককথায় বলতে গেলে গিট ট্যাগ হলো আমাদের প্রজেক্টের কাজের জন্য একেকটি গুরত্তপুর্ন চেকপয়েন্ট। আমরা ট্যাগ দ্বারা বিভিন্ন ভার্সনে মুভ করতে পারি।

**Git Bangla tutorial: GitHub**

‌গিটহাব এমন এক‌টি প্লাটফর্ম যা‌কে আমরা কোড হোস্ট করার কা‌জে ব্যবহার ক‌রি। এখা‌নে আমা‌দের রি‌মোট রি‌পো‌জিট‌রি থা‌কে এবং যে‌কোন সাই‌জের প্র‌জে‌ক্টে টিমওয়ার্ক করার সকল টুলস্ ই আ‌ছে গিটহা‌বে। গিটহা‌বে কোন প্র‌জেক্ট‌কে রি‌পো‌জিট‌রি বলা হ‌য়ে থা‌কে। গিটহা‌বে আমা‌দের কোন রি‌পো‌জিট‌রি বা প্র‌জে‌ক্টে কাজ কর‌তে হ‌লে অবশ্যই এক‌টি গিটহাব একাউন্ট এর দরকার হ‌বে। আমা‌দের কোডবেস য‌দি গিটহা‌বের ম‌তো ক‌োন রি‌মোট রিপো‌জিট‌রি তে থা‌কে ত‌বে নি‌চের সু‌বিধাগু‌লো প‌বো,

1. প্র‌জে‌ক্টের সকল ক‌ন্ট্রিবিউটর এক‌টি কমন রি‌মোট রি‌পো‌জিট‌রি থে‌কে লে‌টেস্ট ভার্সন এর কোড নি‌তে পার‌বে।
2. য‌দি কোন ক‌ন্ট্রি‌বিউটর কোন নতুন ফিচার Debian ভি‌ত্তিক os এর জন্যযুক্ত ক‌রে ত‌বে সেটা এক‌টি রি‌পো‌জিট‌রি তে পুশ ক‌রে রাখ‌লেই হ‌বে। আলাদা ক‌রে সকল ক‌ন্ট্রি‌বিউটর‌কে দি‌তে হ‌বে না।
3. কোলা‌বে‌রেটর এবং ক‌ন্ট্রি‌বিউটররা একসা‌থে গিটহা‌বের অ‌টো‌মেশন টুলস্ ব্যবহার ক‌রে দ্রুততার সা‌থে কাজ কর‌তে পার‌বে।

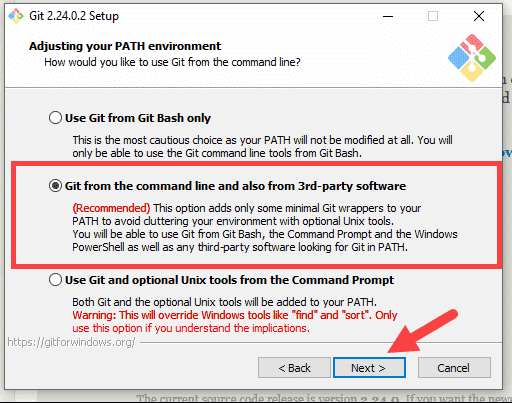
**Git Bangla tutorial: গিট ইন্সটল করা**

**1.1 Debian ভি‌ত্তিক os এর জন্য**

‌Debian ভি‌ত্তিক os যেমন উবুন্টু তে গিট ইন্সটল কর‌তে টা‌র্মিনা‌লে sudo apt install git-all কমান্ড দি‌তে হ‌বে। ইন্সটল হ‌য়ে গে‌লে git --version কমান্ড দি‌লে গিট এর বর্তমান ভার্সন দেখা‌বে।

**1.2 Windows ভিত্তিক OS এর জন্য**

Windows এ গিট ইন্সটল করার জন্য এই লিঙ্ক থেকে গিট ডাউনলোড করে ইন্সটল করে নিন। ইন্সটল করার একধাপে PATH variable setup option আসলে এটা সিলেক্ট করে দিবেন।



এতে করে আপনি CMD (Command line) থেকে গিট কমান্ড দিতে পারবেন। আপনি git bash থেকেও গিট কমান্ড দিতে পারবেন।

গিট ইন্সটল হয়ে গেলে cmd তে কমান্ড দিন, git --version। যদি git version 2.17.1 দেখায় তাহলে ওকে আছে 🙂 ।

**2. প্রথমবার গিট কনফিগার করা**

গিট কনফিগার করার জন্য আমাদের git config কমান্ড দিতে হবে। গিট এর কনফিগ কমান্ড এই লেভেল গুলিতে কাজ করতে পারে,

1. **লোকাল লেভেল (Local level)**: লোকাল লেভেল গিট কনফিগ শুধু প্রজেক্ট ফোল্ডারেই সীমাবদ্ধ থাকে। এর কনফিগারেশন অপশন হলো **–local**। গিটের কমান্ডে যদি কোন কনফিগারেশন অপশন দেয়া না হয় তবে Default ভাবে লোকাল লেভেল কনফিগ করা হয়।
2. **গ্লোবাল লেভেল (Global level)**: গ্লোবাল লেভেল কনফিগ শুধু একটি অপারাটিং সিস্টেমের নির্দিস্ট ইউজারের উপর প্রয়োগ হয়। তার মানে একজন ইউজার আরেকজন ইউজারের কনফিগারেশন এক্সেস করতে পারবে না। এর কনফিগারেশন অপশন প্যারামিটার হলো **–global**.
3. **সিস্টেম লেভেল (System level)**: সিস্টেম লেভেল কনফিগারেশন একটি অপারেটিং সিস্টেমের সকল ইউজার ব্যাবহার করতে পারে। এর অপশন প্যারামিটার হল **–system.**

গিট ইন্সটল করার পরে আমাদেরকে প্রথমেই ইউজার নেম (User name) এবং ইমেইল সেটআপ করে নিতে হবে। এর জন্য আমরা (লিনাক্স) টার্মিনাল এ গিয়ে কমান্ড করবো/ Windows ইউজার রা cmd তে গিয়ে/ git bash চালু করে এই কমান্ড দিতে হবে,

|  |  |
| --- | --- |
| 1  2 | git config --global user.email "sharifhasan@iishanto.com"  git config --global user.name "Sharif Hasan" |

এই কমান্ড করার পরে আমাদের কনফিগ গিটের গ্লোবাল লেভেলে সেভ হবে। –global বাদ দিলে লোকাল লেভেলে সেভ হবে। কনফিগ সেট হয়েছে কিনা তা দেখার জন্য আমরা নিচের কমান্ড দুটি দিবো,

|  |  |
| --- | --- |
| 1  2 | git config user.email  git config user.name |

**Git Bangla tutorial for beginners**

**নতুন প্রজেক্টে গিটের ব্যাবহার**

**1.1 গিটহাবে নতুন গিট একাউন্ট তৈরি করে নিতে হবে**

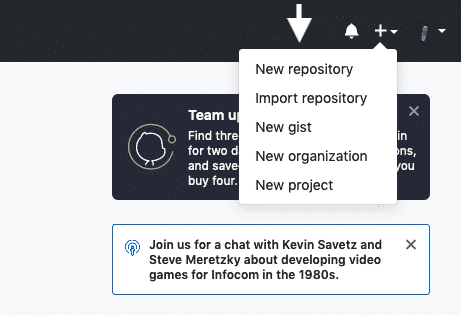
নতুন প্রজেক্টে কাজ করার জন্য প্রথমেই আমাদের গিটহাবে একটি নতুন রিপজিটরি বানাতে হবে। এর জন্য আমাদের গিটহাবে একটি একাউন্ট খুলে ফেলতে হবে প্রথমে। এর জন্য [গিটহাবের মূল পেজে](https://github.com/" \t "_blank) গিয়ে প্রদত্ত ফর্ম টা সঠিক ভাবে পূরণ করতে হবে, এবং সাবমিট করে ভেরিফাই করতে হবে। সব হয়ে গেলে লগইন করে নিতে হবে।

**1.2 গিটহাবে নতুন রিপোজিটরি তৈরি করা**

গিটহাবের (Github) মূল পেজ এ গিয়ে ডান কোনায় + চিহ্নতে ক্লিক করি।

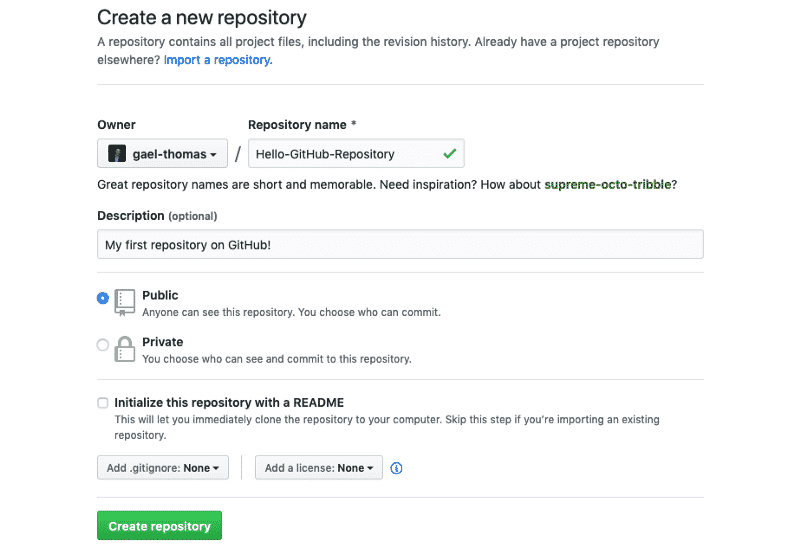
GitHub menu bar with “+” icon

ক্লিক করার পর নিচের মতো মেনু হাজির হবে। এখানে New repository তে ক্লিক করার পর রিপজিটরি তৈরি করার পেজ হাজির হবে।

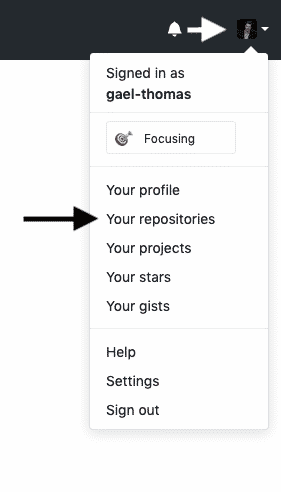


নতুন রিপোজিটোরি বানানোর সাব মেনু

এরপর আমরা রিপোজিটরির (Repository) এর জন্য একটি নাম দিবো এবং Create repository ক্লিক করার আগে ছোট করে একটু Description দিয়ে দিবো। এবং আপাতত আমরা “Initialize this repository with a README” এখানে টিক/মার্ক দিবো না।

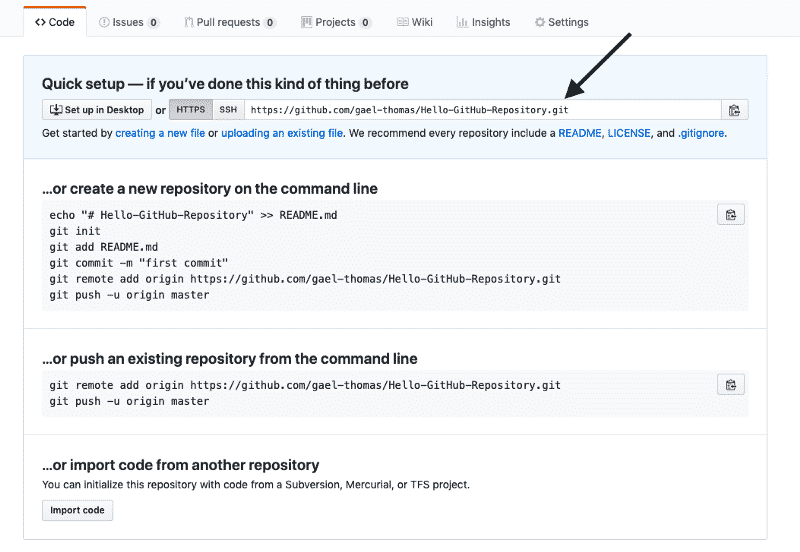
গিট রিপজিটরি তৈরি করা

কাজ শেষ আপনার রিপজিটরি তৈরি হয়ে গেছে। এবার আপনার প্রোফাইল পিকচারের উপর ক্লিক করে > “*Your repositories*” এ ক্লিক করুন। এবং পরের পেজ থেকে আপনার নতুন রিপোতে চলে যান। (git Bangla tutorial)



**2.1 আমাদের গিট রিপজিটরির লোকাল ভার্সন তৈরি করা**

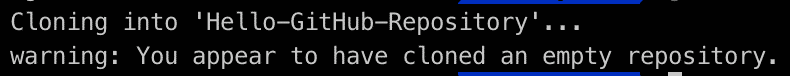
আমাদের প্রথম কাজ হলো, আমাদেরকে আমাদের নতুন রিপোজিটরির একটি অফলাইন কপি তৈরি করতে হবে। যার জন্য আমাদেরকে প্রথমে আমাদের রিপোকে ক্লোন করে নিতে হবে git clone [https\_address] এই কমান্ড ব্যাবহার করে। আপনার নতুন রিপোর হোম পেজে নিচের মতো দেখাবে। সেখান থেকে https টা আমরা কপি করে নিবো।

গিট রিপো

কপি করার পরের ধাপে আপনি আপনার কম্পিউটারের CMD/Terminal/Git Bash চালু করে নিচের কমান্ড লিখুন

|  |  |
| --- | --- |
| 1 | git clone সদ্য\_কপি\_করা\_https\_address\_এখানে |

এর আউটপুট নিচের মতো হবে,

Git cloning into empty repository

এখন আমরা cd আমাদের\_রিপোজিটরির\_নাম কমান্ডের মাধ্যমে cmd/terminal/git bash এর লোকেশন চেঞ্জ করে আমাদের রিপোতে নিয়ে যাবো। আপনি আপনার প্রজেক্টকে যেখানে ক্লোন করেছেন (Terminal/Cmd এর প্রতি লাইনেই যেখানে বর্তমান লোকেশন তা দেখায় ie. C:\users\your name etc. ) সেখানে দেখতে পাবেন।

**2.2 নতুন ফাইল তৈরি করা**

এবার আপনার ফাইল ম্যানেজার দিয়ে আপনার যেখানে রিপো তৈরি হয়েছে সেখানে “*README.md*” নামে একটি নতুন ফাইল তৈরি করুন এবং notepad দিয়ে edit করে নিচের হাবিজাবি মার্কডাউন দিয়ে ভরে ফেলুন (ইচ্ছা মতো পরিবর্তন করা যাবে)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | ### My first repository is awesome!  I'm [PSEUDO/NAME] and here it's my first GitHub repository.  If you see this file on my project, it's because I'm learning Git.  My mood:  > [MOOD NAME]  My favorite color:  > [COLOR NAME]  Thank you so much for reading! ☺ |

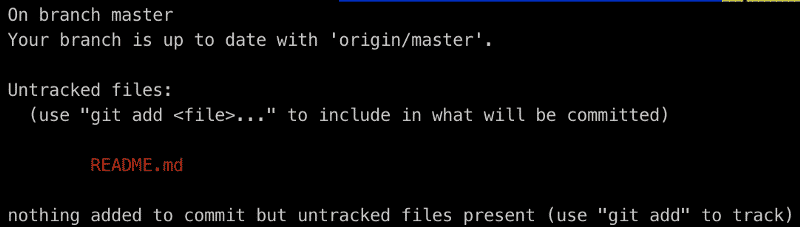
**3.1 আমাদের মডিফাইড প্রজেক্টকে পুনরায় গিট রিপোজিটরি তে সেভ করা**

এখন আমরা আমাদের তৈরি করা প্রজেক্টকে মডিফাই করেছি। তো এখন আমাদের এটা অনলাইন রিপজিটরিতে সেভ করতে হবে। এই প্রসেসের নাম হলো কমিটিং (Committing). এটা করার জন্য আমরা আবার cmd/terminal/git bash এ ফিরে যাবো (আপনি এই window কেটে দিলে পুনরায় cd কমান্ড ব্যাবহার করে প্রজেক্ট ফোল্ডারে চলে যান)

এখন সেভ করার জন্য আমাদেরকে ৪ টি ধাপে আগাতে হবে, “*status*”, “*add*”, “*commit*” এবং “*push*”। এই প্রত্যেকটা অপারেশনের সময় আপানার cmd/terminal/git bash এর লোকেশন প্রজেক্ট ফোল্ডারের ভিতরে নিয়ে নিতে হবে।

**git status command**

আপনি git statusকমান্ড দিলে টার্মিনালে যেসব ফাইল মডিফাই করা হয়েছে/ তৈরি করা হয়েছে তার একটা লিস্ট পাবেন। নিচের ছবির মতো।

“git status” output in our project

**git add command**

git add কমান্ড দিলে আমাদের নির্দিস্ট ফাইলকে স্টেজিং এরিয়া (Staging area) তে নিয়ে যাওয়া যাবে। git add এর সিনট্যাক্স হলো

|  |  |
| --- | --- |
| 1 | $ git add [FILENAME] [FILENAME] [...] |

আমরা আমাদের README.md ফাইলটা যুক্ত করবো নিচের কমান্ড দিয়ে,

|  |  |
| --- | --- |
| 1 | $ git add README.md |

আমরা যদি git add . কমান্ড দেই তবে আমাদের সকল ফাইল একসাথে Staging area তে চলে যাবে। [ প্যাঁরা নাই, চিল ]। এখন যদি আমরা git statusকমান্ড আবার দেই তবে আমাদের README.md ফাইল সবুজ দেখাবে।

**git commit command**

এখন আমরা আমাদের পছন্দ মতো ফাইলগুলো add করে নিয়েছি। এবার আমাদের একটি মেসেস দিতে হবে, যে আমরা আসলে কি করেছি। আপনি যদি পরবর্তিতে “Change history” চেক করতে যান তাহলে বিষয়টা হেল্পফুল হবে। আমরা যা কমিট করবো তার কমান্ড নিচের মতো।

|  |  |
| --- | --- |
| 1 | $ git commit -m "Added README.md with good description in it." |

**git push command**

আমাদের লোকালের কাজ শেষ। এবার আমরা আমাদের প্রজেক্টকে অনলাইনে সেভ করবো। এর জন্য আমাদের গিট পুশ কমান্ড ব্যাবহার করতে হবে। যার সিনট্যাক্স নিচের মতো

|  |  |
| --- | --- |
| 1 | $ git push [remote name] [branch name] |

আমাদের প্রজেক্টকে পুশ করতে হলে নিচের মতো কমান্ড করবো,

|  |  |
| --- | --- |
| 1 | $ git push origin master |
|  |  |

আমাদের কাজ শেষ। এখন আমরা যদি আমাদের github repository তে আবার যাই তাহলে নিচের মতো দেখতে পাবো।

