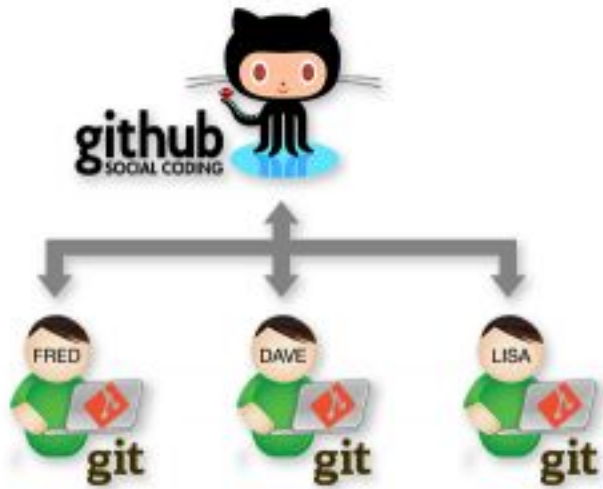


ওপেন সোর্স এ contribution করার জন্য [Github](#) এর কোন বিকল্প খুব কমই আছে। **Github** খুব সম্ভবত সবচেয়ে বড়ো কোড hosting সাইট। github এ পাবলিক অথবা প্রাইভেট (সামান্য অর্থের বিনিময়ে) repository খোলা যায়।



### যে জন্য আপনি ওপেন সোর্সে আপনার শ্রম দিবেন

কারণটা হচ্ছে, আপনার নিজের ভালোর জন্য, আপনি ওপেন সোর্সে আপনার যে স্কিল, শ্রম, এবং সময় দিবেন সেটা আসলে আপনার নিজেকে আরও দক্ষ করা জন্য, একটা উদাহরণ দেই ধরা যাক আপনি [nodejs](#) এর একটা বাগ খুঁজে বের করলেন এবং pull request সেন্ড করলেন তখন nodejs এর contributor list এ আপনার নামটা উঠে যাবে, বাগটা যেহেতু আপনি খুঁজে বের করেছেন সেহেতু আপনার ডিবাগিং স্কিলটা ঝালাই হয়ে গেলো এছাড়া আপনার জব পাওয়ার জন্য ওপেন সোর্স contributor রা অগ্রাধিকার পায়। গুগল, ফেসবুক থেকে শুরু করে মাইক্রোসফট পর্যন্ত এখন ওপেন সোর্সের দিকে ঝুঁকছে। [এই লিঙ্কে](#) পাবেন ফেসবুক এর টপ স্টারড ওপেন সোর্স প্রোজেক্ট। সবচেয়ে বড় কথা আপনি শিখতে পারবেন।

অনেক ওপেন সোর্স প্রোজেক্ট গিটহাবে তাদের বিশাল community নিয়ে কাজ চালিয়ে যাচ্ছে। যে কেউ পাবলিক repository এর কোড পড়তে পারে এর জন্য কোন github অ্যাকাউন্ট প্রয়োজন হয় না, এমনকি মাউসের একটি click এ সম্পূর্ণ repository নিজের পিসিতে ক্লোন করা যায়।

গিটহাবে repository খোলার জন্য আপনার একটি গিটহাব অ্যাকাউন্ট লাগবে তারপর public অথবা private repository খুলতে পারবেন। নতুন repository খোলার সময় README এবং ওপেন সোর্স লাইসেন্স দিয়ে repository কে initialize করতে হয়।



## Readme

ফাইলে আপনার প্রোজেক্ট সম্পর্কে বিস্তারিত লিখতে হয় যাতে সবাই পড়ে বুঝতে পারে প্রোজেক্টটি সম্পর্কে এবং যদি কেউ চায় তাহলে contribute করতে পারে।

আমি এই লেখায় গিটহাবে ব্যবহার করা হয় এমন কিছু টার্ম নিয়ে আলোচনা করবো কেননা গিটহাবে কোন প্রোজেক্টে contribute করতে গেলে বা নিজের প্রোজেক্ট খোলার জন্য এই টার্ম গুলো জানা থাকতে হবে, যেমনঃ repository, pull request, wiki , issues, commit, branch, commit, fork.

## Repository

কোন নতুন প্রোজেক্ট গিটহাবে অ্যাড করার জন্য প্রথমে ওই প্রোজেক্টের একটা repository খুলতে হবে। repository তে প্রোজেক্টের সব ফাইল, ইমেজ, সহ বিস্তারিত থাকবে। এই repository হতে পারে লোকাল কম্পিউটারে অথবা কোন ওয়েব সার্ভারে।

repository তে একটি readme file থাকা উচিত যেখানে প্রোজেক্টটি সম্পর্কে বিস্তারিত বর্ণনা করা থাকবে।

## Branch

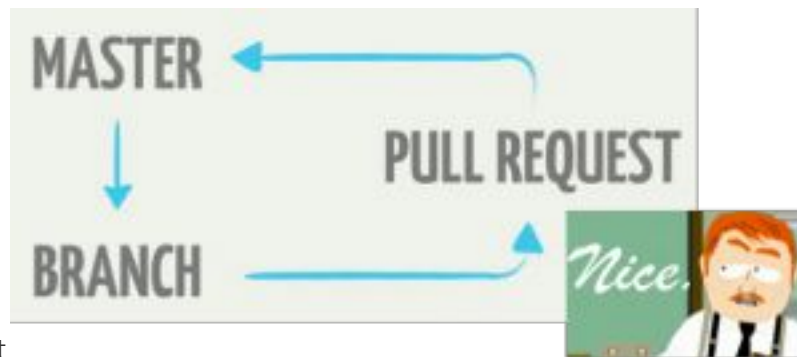
Github এ যখন একটি প্রোজেক্ট খোলা হয় তখন একটা master branch ক্রিয়েট হয়ে যায়। ওই master বা প্যারেন্ট branch এর সাথে প্রোজেক্ট এর সব ফাইল বা directory অ্যাড করতে হয়।

ধরা যাক , আপনি কোন প্রোজেক্টের কোন branch এডিট করতে চান তাহলে আপনি একটি নতুন branch create করে প্রয়োজনীয় এডিট করে branch টি master branch এর সাথে merge করে দিবেন। এটা অনেকটা ট্রি এর মতো - master branch রুট এবং সাবডিরেক্টরি / ফাইল গুলো চাইল্ড।



## Fork

ধরা যাক আপনি অন্য কোন প্রজেক্টের এ নিজের repository তে কপি করতে চান তাহলে আপনাকে প্রজেক্টটি fork করতে হবে। তারপর আপনি যদি চান ওই প্রজেক্টে এর কোন branch এডিট করতে বা কোন বাগ ফিক্স করতে তাহলে সেটা করে আপনাকে একটি pull request পাঠাতে হবে original author এর কাছে এভাবে আপনি কোন প্রজেক্টে contribute করতে পারেন।



## Pull Request

ধরা যাক আপনি কোন ওপেন সোর্স প্রজেক্টে contribute করতে চান, আপনি ওই প্রজেক্টে একটি বাগ খুঁজে পেয়েছেন এবং বাগটি ফিক্স করেছেন তখন আপনাকে একটি pull request বিস্তারিত বর্ণনা সহ পাঠাতে হবে ওই প্রজেক্ট যে maintain করছে তার কাছে তখন সে ওই pull request টা দেখে যদি accept করে তাহলে অ্যাড করে দিবে (ignore ও করতে পারে যদি উল্টাপাল্টা request পাঠানো হয়) মূল repository তে। বেশিরভাগ সময় pull request অটোম্যাটিকালি প্রজেক্টে অ্যাড হয়ে যায় কিছু ক্ষেত্রে maintainer কে ম্যানুয়ালি কিছু pull-request marge করতে হয়।

৩টি ধাপ রয়েছে কোন ওপেন সোর্স প্রোজেক্টে contribute করার জন্য।

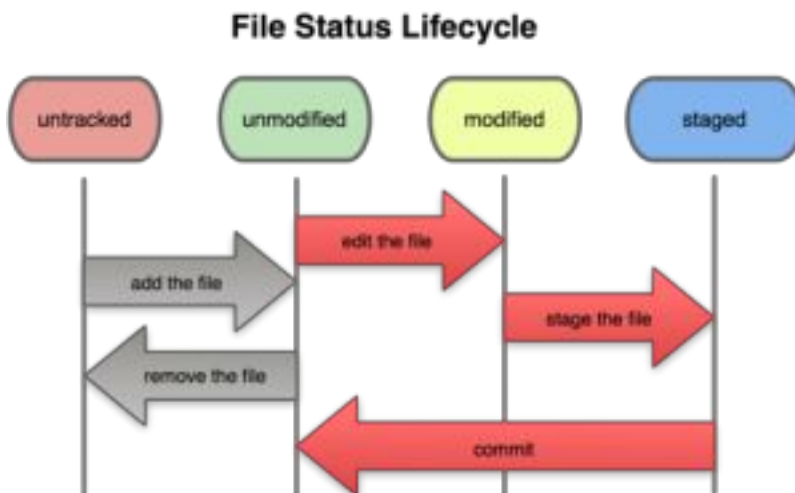
1. **Fork the project**
2. **Create branch**
3. **Send Pull request.**

## Push

লোকাল পিসি থেকে কোন repository কে ওয়েব repository তে upload করার জন্য push করতে হয়।

## Issues

কোন প্রোজেক্টের যদি বাগ খুঁজে পান কোথাও কোন প্রবলেম থাকলে ওই বিষয়ে issue create করতে হয়, তখন ওই প্রোজেক্ট যারা ফলো করতেছে বা contribute করতেছে অথবা project maintainer ওই ইস্যুর উত্তর দেয়, আলোচনা করে। এটাকে bug tracker হিসেবে কল্পনা করা যায়।



## Commit

যখন কোন repository এর কোন অংশ পরিবর্তন বা এডিট করা হয় তখন সেটা commit করতে হয়। commit করার সময় summary এবং description দিতে হয় যে কি change করা হল। commit করা মানে হল পরিবর্তনগুলো সেভ করা। commit না করে কোন Branch পুশ করতে পারবেন না।

এখন আপনি আপনার পছন্দ মতো কোন প্রোজেক্টে contributor হিসেবে কাজ শুরু করে দিতে পারেন। গিটহাবে গিয়েও খুঁজে নিতে পারেন অথবা [codetriage](https://www.codetriage.com/) এই সাইটে গিয়ে পছন্দ মতো প্রোজেক্ট খুঁজে নিতে পারেন খুব সহজেই, যেমন ধরা যাক আপনি শুধু সি বা জাভা পারেন তাহলে আপনি এই সাইটে গিয়ে সি, জাভা রিলেটেড প্রোজেক্ট পেয়ে যাবেন সেখানে কারা contribute করছে সহ আরও তথ্য সহজেই পেয়ে যাবেন।

এছাড়া আপনি আপনার সিভিতে আপনার github রাফিং এর লিঙ্ক অ্যাড করে দিতে পারেন এই লিঙ্ক থেকে [github-awards](#) এই লিঙ্কে গ্লোবালি এবং লোকালি আপনার রাফিং পাওয়া যাবে। আবার আপনি চাইলে আপনার গিটহাব resume ও আপনার সিভিতে অ্যাড করে দিতে পারেন এই [resume.github.io](#) এই লিঙ্ক থেকে , লিঙ্ক গিয়ে আপনার গিটহাব আইডি দিলেই আপনাকে resume generate করে দিবে।  
আজ এই পর্যন্তই ধন্যবাদ সবাইকে।

© copyright reserved by the **Author**.



source : <http://nanotouchbd.com/bn>

Author : Shohanur Rahaman