**1. 깃허브의 역사**

GitHub.com 플랫폼은 2007년 10월 19일에 만들기 시작했다. 2008년 4월에 톰 프레스턴-워너, 크리스 완스트라스, P. J. 하이엣, 스콧 샤콘이 베타 버전으로 몇 달 동안 사용 가능하게 만든 후, 이 사이트를 시작했다.

**1-1. 조직구조**

원래 GitHub, Inc.는 임원이 없는 수평적인 조직이다. 즉 모든 사람은 관리자라 할 수 있다. 직원들은 관심있는 프로젝트를 선택하고 급여만 최고 경영자에 의해 정해졌다. 그러다 2014년 GitHub, Inc.는 중간 관리직을 도입했다.

**1-2. 경영**

GitHub.com은 창업 1년차에는 3명의 창업자로도 자금을 조달하고 직원을 인수할 수 있는 충분한 수익이 가능했다. 회사가 설립된 지 4년 후인 2012년 7월, Andreessen Horowitz는 벤처 캐피털에 1억 달러를 투자받았다. 2015년 7월 GitHub는 시리즈 B 라운드에서 2억 5천만 달러의 벤처 자금을 추가로 조달했고 투자자는 세쿼이아 캐피털, 안드레센 호로위츠, 스트로우 캐피털 등 벤처 캐피털이었다. 2016년 8월 현재 GitHub은 연간 1억 4천만 달러의 수익을 올리고 있다.[8]

**2. 깃허브란?**

우선, Github에서 Git의 뜻을 알아보자. Git은 형상 관리 도구 중 하나로, 컴퓨터 파일의 변경사항을 추적하고 여러 명의 사용자들 간에 해당 파일들의 작업을 조율하기 위한 분산 버전 관리 시스템이다. Git은 소프트웨어 개발에서 소스 코드를 효과적으로 관리할 수 있도록 하는 무료로 하는 공개 소프트웨어이다. Git은 프로젝트 폴더 내에서 작업을 기록한다. 이런 Git의 장점을 정리해보면, 소스코드를 따로 주고 받을 필요 없이, git을 사용하면 하나의 프로젝트, 같은 파일을 여러 사람이 동시에 작업하는 병렬 개발이 가능하다는 것이다.

이를 알고 Github의 뜻을 알아보자. Github란, 컴퓨터 프로그램 소스를 공유하고 협업하여 개발할 수 있는 버전 관리 시스템인 깃(Git)에 프로젝트 관리 지원 기능을 확장하여 제공하는 웹 호스팅 서비스이다. 앞서 말했듯 소프트웨어 소스 코드를 공유하고 협력하면서 개발할 수 있도록 지원하는 분산형 버전 관리 시스템으로, 코드 수정 권한, 버전 추적 등 소프트웨어 개발에 필요한 관리 기능을 제공한다. 분산형 버전 관리 시스템(DVCS)은 중앙 집중 방식의 문제를 해결한다. 공동 버전 시스템(CVS: Concurrent Versions System), 서브버전(Subversion), 퍼포스(Perforce) 등과 같은 중앙 집중 방식의 버전 관리 시스템(CVCS: Central Version Control System)은 중앙 서버에서 모든 파일의 버전을 저장·관리하고 다수 클라이언트가 서버에서 파일들을 가져와서 사용하는 방식으로 서버에 장애가 발생하면 사용을 할 수 없고 데이터 복구도 할 수 없다.이와 달리 DVCS에서는 클라이언트가 저장소를 통째로 복제하여 사용하기 때문에 서버에 문제가 발생해도 클라이언트는 복제된 저장소를 다시 서버에 복사하여 서버 내 데이터를 복원할 수 있다. 깃허브(GitHub)는 깃(Git)의 기본 기능을 포함하여 프로젝트 관리에 필요한 버그 추적(bug tracking), 기능 요청(feature requests), 작업 관리(task management), 위키(wiki) 기능 등을 추가적으로 제공한다.

마지막으로 깃과 깃허브의 차이점은, 깃(Git)이 텍스트 명령어 입력 방식인데 깃허브는 화려한 그래픽 유저 인터페이스(GUI)를 제공한다.

**3. 깃허브의 장점**

깃허브의 첫번째 장점으로는, 깃의 장점을 그대로 가지고 있다. 모든 변경사항에 대한 “스냅샷”을 저장하기 때문에 어떠한 데이터도 잃어버리지 않는다. 또한 개발자의 소셜네트워크로 묶인 공간이기 때문에 소스가 버저닝되어 조금 더 좋은 소스로 업데이트 하기위한 공간이기도 하다. 또한 효과적인 협업도 가능하다라는 부분을 설명할 수 있다. 다시 말 해, 잘못된 코드도 소셜로 누군가의 도움을 받을 수 있다는 말입니다. 두번째 장점으로는 효율적인 배포관리이다. 깃허브를 이용하면 버전관리, 개발용, 운영용, 그리고 private 영역으로 별도 구성을 할 수 있다. 마지막으로는 여러 명이 병렬적으로 개발 할 수 있게 해주어 개발 속도를 매우 빠르게 해준다. 즉, 동시에 프로젝트 작업이 가능하다는 것이다.

**4. Github 및 오픈 소스 코드의 엔드리스 저장소에 대해 알아야 할 사항**

먼저 Git과 GitHub는 동일하지 않다는 점. "Github"의 "Git"은 Global Information Tracker를 의미한다. Linux에서 작업하는 개발자를 돕기 위해 2005년에 만들어진 소스 코드 기록을 관리하고 추적하는 오픈 소스 버전 제어 시스템이다.

GitHub는 Git 아카이브를 보다 쉽게 관리하는 것을 목표로 하는 클라우드 기반 호스팅 서비스이자 회사이다. 특히 오픈 소스 코드 프로젝트를 찾고 작업하는 개발자에게 굉장히 유용하다. Windows 또는 Safari에서 git 의 설치가 중단되는 유일한 회사는 아니다. 그러나 일부 사람들은 왜 GitHub 오픈 소스 자체 코드를 열지 않는지 궁금해했다. GitHub는 이 질문에 대한 답을 얻을 수 있는 회사이다. GitHub에서 팀은 다른 프로그래머가 자신의 컴퓨터에 복사할 수 있는 저장소의 마스터 복사본을 업로드한다.이러한 변경 사항은 결국 마스터 복사본 또는 저장소와 병합된다.

**5. Github에서 기본적으로 알아야 할 것**

저장소(Git Repository) : 소스코드가 저장되어 있는 여러 개의 브랜치들이 모여 있는 디스크상의 물리적 공간을 의미한다.

커밋(commit) : 파일을 추가하거나 변경 내용을 저장소에 저장하는 작업

푸시(push) : 파일을 추가하거나 변경한 내용을 원격 저장소에 업로드하는 작업

저장소는 내 컴퓨터에 있는 로컬 저장소, 서버 등 네트워크에 있는 원격 저장소 2개가 있다.

로컬(local)저장소 : 현재 내가 사용하고 있는 내 PC에 저장되는 저장소이며, 내 PC의 폴더와 유사하다. 원격 저장소의 파일들을 로컬저장소로 pull할 수 있다.

원격(Remote) 저장소 : 원격 저장소는 원격 서버에 저장되고 관리되는 저장소이며, 로컬 저장소의 파일들을 원격 저장소에 push할 수 있다.

working directory파일은 크게 tracked, untracked의 두 가지 상태로 나뉜다. 파일의 수정이 일어나면 git이 파일의 변경을 감지하여 사용자에게 알려주는, 파일을 추적하는 tracked상태와 파일을 저장소에 저장할 필요가 없어 git이 추적하고 있지 않은 untracked상태이다.

[ 출처 ]

https://physicallaw.tistory.com/135

https://developer.oracle.com/ko/learn/technical-articles/github#chapter-github%EC%9D%84-%EC%82%AC%EC%9A%A9%ED%95%98%EB%8A%94-%EC%9E%A5%EC%A0%90%EA%B3%BC-%EB%8B%A8%EC%A0%90

https://ko.wikipedia.org/wiki/%EA%B9%83%ED%97%88%EB%B8%8C

https://chae528.tistory.com/38?category=987163

#include <studio.h>

int main(void)

{

int num1 = 10, num2 = 12;

int result1, result2, result3;

result1 = (num1 == 10 && num2 == 12);

result2 = (num1 < 10 || num2 > 10);

result3 = (!num1);

printf("result1: %d \n" , result1);

printf("result2: %d \n" , result2);

printf("result3: %d \n" , result3);

}