GitHub는 미국의 대표적인 무료 Git 저장소이다. GitHub는 4명의 창업자가 만든 무료 Git 저장소로, 2008년에 출시되었다. 여기서 Git는 리누스가 개발한 분산형 버전 관리 시스템이다. ‘버전’이란 프로그램의 변경사항을 말하는데, Git을 사용하면 프로그램을 변경할 시, 그 이력을 확인하고 관리할 수 있다. GitHub를 사용하기 전, 몇 가지 우리에게 익숙하지 않은 용어들이 있는데, 먼저 Repository는 파일들이 저장되는 장소이다. 즉, 프로젝트들의 폴더이다. Repository의 종류는 크게 두 가지로 나뉘는데, Remote Repository (원격 저장소)와 Local Repository (개인저장소)로 나뉜다. GitHub Repository 기능은 GitHub의 대표적인 서비스로, 프로그램 소스코드를 저장케 하고 개발자가 사용할 수 있는 기능이다. Commit은 프로그램의 변경이력을 뜻하고, Stage는 Commit을 통해 변경사항들이 반영되기 전에 해당 변경사항들의 내용들이 저장되는 공간을 말한다. 특정 파일이나 코드를 수정할 시, 이 이력들이 Stage area에 기록되며, 이 행위를 Staging이라고 한다. Branch는 특정 Commit으로부터 분기되는 환경을 말한다. 즉, 각 개발자들이 개발을 진행하고 있는 환경이다. Merge는 나누어진 Branch를 다시 하나의 branch로 합치는 것을 의미한다.

Git의 체제를 정리하자면, 일단 Git은 Local Repository 라는 이름으로 사용자의 데이터를 작업 폴더에 넣어 관리한다. 이 Local Repository는 해당 작업자의 작업을 버전 별로 저장하고 관리하며, 동시에 다른 작업자의 작업을 다른 Local Repository로 다원화 하는 역할을 한다. 작업이 끝나면 Git의 Remote Repository로 다시 발행된다

Git의 장점으로는, 일단. 속도가 매우 빠르다. 왜냐하면 각각의 개발자들이 모두 분산 처리된 서버의 주인이 되는 것이므로 서버 자체가 해야 할 일은 많이 줄어들기 때문이다. 또한, 일시적인 서버 장애가 발생하더라도, 로컬 저장소를 이용하면 되기 때문에 개발을 계속할 수 있다. 그리고 가지치기 즉, Branch가 가볍고, Merge가 용이하며, Staging의 기능도 지원한다.

이러한 장점 외에도 GitHub는 여러가지 기능과 서비스를 지원한다. 예로, GitHub Advanced Security는 코드 패키지의 취약점과 소스코드의 문제점을 찾아주는 등의 보안 기능을 하고, GitHub Pages는 간단한 마크다운(Markdown) 형식 파일을 GitHub Repository에 저장 및 설정을 하면 웹사이트를 만들어주는 기능이다. 그리고 GitHub Copilot은 인공지능과 머신러닝을 활용하여 개발자가 적는 코드의 내용을 자동으로 파악하여 나머지를 대신 적어주는 혁신적인 기능이다.

하지만 단점 또한 존재하는데, 기존의 형상관리 도구에 비해 초보자에게는 배우기가 어렵다. 기존의 중앙집중형 도구에 비해, Git은 작업 디렉토리-스테이지-로컬 저장소-원격 저장소의 다층구조로 되어있다. 각각의 계층들은 단순히 반복되는 것이 아니라, 훨씬 복잡하게 연결되어 있다. 또한 여러 어려운 용어들이 존재하여 기존의 초보자가 갖고 있던 지식으로는 이 용어들의 개념을 완벽하게 이해하기가 쉽지 않다. 이러한 단점에도 불구하고, Git은 개발 상황에서 굉장히 유용한 장점들이 많기 때문에 오늘 날 GitHub는 오픈 소스로써 개발자들에게 사랑받고 있다.

이제 본격적으로 GitHub의 사용법을 알아보자. 먼저 검색창에 GitHub를 검색하여, GitHub에 접속한다. GitHub에 회원가입이 안되어 있다면 회원가입을 하여 로그인을 한다. 그 이후 새로운 저장소를 생성하기 위해 ‘New’ 라는 버튼을 클릭한다. 프로젝트의 이름을 입력하고, 프로젝트를 전체공개 할 것이며 ‘Public’을, 비공개로 할 것이면 ‘Private’를 클릭한다. 프로젝트를 전체 공개하여야 다른 개발자들이 나의 코드를 볼 수 있다. 이렇게 저장소를 생성한 후, Git을 사용하기 위해 Git을 다운받아야 한다. Git을 다운받은 후 Git bash를 열어 환경설정을 해줘야 한다. 이 후, 유저네임, 이메일 등 자신의 정보를 입력해준다. 이렇게 Git의 환경설정은 마무리한다.

이렇게 GitHub를 이용하기 위한 설정을 세팅한 후 프로그램을 올리기 위해 미리 짜둔 코드 창에서 ‘Terminal’을 클릭한다. 이 후 ‘Git init’ 부터 시작하여 코드를 작성한다. 마지막으로, ‘Git push origin master’로 master branch로 보내면 우리가 짠 코드를 성공적으로 GitHub로 보낼 수 있다.

GitHub는 한국을 포함하여 전세계 IT 기업에서 취업자들에게 요구하는 포트폴리오이다. 아예 입사지원서에 GitHub 계정을 요구하는 기업도 많아지고 있는 추세이며, 입사 과정에서뿐만 아니라, 스타트업이나 개방적인 IT기업에서는 GitHub을 이용하여 기업 활동에 사용하는 일이 점점 늘고 있다. 그러므로 프로그래머를 꿈꾸는 학생들은 코드를 작성하는 능력 외에도 GitHub에 대한 이해도와 GitHub를 사용하는 방법을 알 필요성이 있다.