

GitHub에 관하여

이번 활동을 통해 저는 GitHub의 역사, 기능 그리고 사용법에 대해 찾아보게 되었습니다. 먼저 역사에 대해 설명 드리겠습니다. 깃허브의 시작은 2007년 10월 19일에 톰 프레스턴위너와 크리스 완스트래스, P. J.하이엣가 깃의 문제점 해결을 위해 만들기 시작했습니다. 깃허브는 그렇게 2008년 4월부터 사용이 가능하게 되었고 임원이 없는 수평 구조를 이루며 문제 없이 운영되다가 2016년 6600만 달러의 적자를 기록하게 되었습니다. 그때 깃허브는 인수되는 것을 희망했고 마이크로소프트의 사티아 나델라에 의해 마이크로소프트사로 2018년 6월 4일에 인수되었습니다. 많은 사람들이 마이크로소프트에 인수되면서 깃허브가 더 이상 오픈 소스를 공개하지 않고 배척하게 될까봐 걱정했던 것과 다르게 오픈 소스를 배척하지 않았습니다. 하지만 마이크로소프트사가 인수했기에 마이크로소프트사와 경쟁관계에 있는 애플, 구글등의 많은 기업들은 조금 불편해졌다고 전해졌습니다. 여기까지가 역사에 관련된 이야기였고 지금부터 제가 직접 사용해보거나 인터넷에서 찾아보면서 생각한 GitHub에 대해 설명드리겠습니다. 제가 GitHub에 대해 찾아보고 사용해본 결과 GitHub의 주요한 기능은 코드 저장, 사람들끼리의 코드 공유 두가지로 정리할 수 있었습니다. 지금부터 제가 알아본 GitHub 사용법에 대해 말씀드리겠습니다. 첫째, 코드를 저장할 수 있다. 제가 알아본 바로는 GitHub에는 코드를 저장할 수 있습니다. 어떻게 코드를 저장 가능하냐면 자신의 계정에 저장소를 만들어 그곳에 코드들을 저장해놓는 것입니다. 일단 저장소를 만들어 코드를 저장하는 방법에 대해 설명드리겠습니다. 먼저 GitHub를 들어간 뒤 오른쪽 상단을 살펴보면 프로필 옆에 +표시가 있습니다. +표시를 누르게 된다면 "New repository" 라는 항목이 보입니다. 이 항목을 누르게 된다면 자신의 Repository 즉 저장소를 만들 때 세부적으로 설정 가능한 화면이 뜨게됩니다. 화면에 들어가게 된다면 먼저 저장소가 자신만의 것인지 혹은 자신이 속한 단체의 것인지를 정하고 그 옆에 "Repository name" 즉 저장소 이름을 정해줍니다. 그 후에는 공개 여부를 설정할 수 있는 데 Public으로 설정하면 누구나 볼 수 있게 공개된 저장소가 되는 것이고 Private로 설정하면 자신만 볼 수 있게 됩니다. 그리고 넘어가 마지막으로 저장소를 초기화 할때 생성할 파일을 선택 가능하게 됩니다. 선택 가능한 파일의 종류는 세가지로 "README 파일 추가", ".Gitignore 파일 추가", "license 파일 추가"가 있습니다. 세가지 항목에 대해 자세히 설명해드리겠습니다. 먼저 README 파일 추가의 경우에서 설명하자면 우리는 저장소를 소개하는 문서를 관습적으로 README 라는 이름을 사용하여 나타내고 있으며 포매팅에 따라서 확장자를 지정할 수 있습니다. 따라서 저장소를 생성할 때 이 옵션을 선택하면 GitHub에서는 저장소를 소개하는 말을 저장할 수 있도록 저장소

이름만 쓰여진 README.md 파일을 추가해줍니다. 그 다음으로 .Gitignore 파일 추가의 경우에 대해서 설명드리기에 앞서 .gitignore 파일에 대해 설명드리자면 Git 저장소에서 작업할 때 무시하고 싶은 특정 이름이나 패턴을 나열하는 특수한 파일입니다. 따라서 만약 .Gitignore 파일 추가를 선택한다면 특정한 방법을 실행하지 않는 이상 저장소에 .Gitignore 파일에 추가된 파일은 더 이상 우리가 편집할 수 없게 됩니다. 마지막으로 세번째인 license 파일 추가에 대해 설명하자면 저장소에 라이선스 파일을 추가해줍니다. 따라서 오픈소스 저장소를 만드는 경우 손쉽게 오픈소스 라이선스를 추가할 수 있게 해줍니다. 이로써 3가지 항목에 대해 설명 드렸지만 인터넷에서 찾아본 결과 이 항목들은 이미 로컬에서 초기화한 Git 저장소를 GitHub에 올리려고 하는 경우에 충돌이 발생할 수 있으므로 이 옵션을 선택하지 않는 것을 추천한다고 했습니다. 이렇게 Repository를 생성했다면 어떻게 코드를 저장하는 지를 알려드리겠습니다. 코드를 저장하는 방법에는 직접 GitHub Repository에 코드를 저장하는 방법과 원격 저장소를 이용하여 저장하는 방법으로 두가지 방법이 있습니다. 직접 GitHub Repository에 저장하는 방법은 위에서 말씀드렸듯이 오른쪽 상단을 보면 프로필이 있는데 프로필에 들어가줍니다. 프로필에 들어왔다면 여러가지의 옵션이 존재하는 데 그것들 중 Repositories를 선택해줍니다.

Repositories를 선택했다면 자신이 만든 저장소들이 뜨는데 그것들 중 코드를 저장할 저장소를 고른 뒤에 add file을 눌러줍니다. add file을 눌러주게 되면 Create new file과 Upload files이 나타나게됩니다. 이때 Create new file을 눌러준다면 코드를 입력할 수 있는 창이 뜨게 되는데 자신의 이름 옆에 파일의 이름을 적어주고 그 밑에 자신이 작성한 코드를 눌러준 뒤 Commit Changes를 눌러주면 됩니다. 그리고 만약 Upload files를 눌러준다면 화면의 중심에 "Drag files here to add them to your repository"가 뜨게 되는데 직접 이곳에 저장하고 싶은 파일을 드래그를 하거나 밑의 "Choose your files"를 눌러 자신의 컴퓨터에서 파일을 선택한 뒤 "Commit changes"를 누르면 파일을 저장할 수 있게됩니다. 여기까지가 직접 GitHub Repository에 코드를 직접 저장하는 방법이었고 원격 저장소를 통해 코드를 저장하는 방법은 잘 모르기에 Chat gpt를 통해 찾아보니 로컬 컴퓨터에 Git을 설치하고 설정합니다. Git 설치해 해당 운영체제에 맞는 Git 공식 웹사이트(<https://git-scm.com/>)에서 다운로드를 받을 수 있습니다. 그 이후에는 코드를 저장할 로컬 프로젝트 폴더를 생성하고 해당 폴더로 이동합니다. 명령 프롬프트(Windows) 또는 터미널(Mac/Linux)에서 다음 명령어를 사용하여 Git 저장소를 초기화합니다. 초기화 한 뒤에는 깃허브 원격 저장소와 로컬 저장소를 연결하고 깃허브 저장소 URL을 사용하여 다음 명령어를 실행합니다. 그리고 나서 로컬 프로젝트에 코드를 추가하고 커밋합니다. 새 파일을 만들거나 기존 파일을 수정한 후, 다음 명령어를 사용하여 변경 사항을 스테이징하고 커밋합니다. 로컬에서 커밋한 변경 사항을 깃허브 원격 저장소에 푸시합니다." 라는

절차를 통해 원격 저장소를 사용하여 코드를 저장 할 수 있다고 했습니다. 여기까지가
제가 생각한 GitHub의 주요한 기능에 대한 정보였습니다.