깃허브(github)란 컴퓨터 프로그램 소스를 공유하고 협업하여 개발할 수 있는 버전 관리 시스템인 깃(Git)에 프로젝트 관리 지원 기능을 확장하여 제공하는 웹 호스팅 서비스입니다. GitHub 흐름은 간단한 분기 기반 워크플로입니다. GitHub 흐름은 개발자뿐만 아니라 모든 사용자에게 유용합니다. GitHub 흐름을 따르려면 GitHub 계정 및 리포지토리가 필요합니다. 리포지토리는 일반적으로 단일 프로젝트를 구성하는 데 사용됩니다. 리포지토리에는 프로젝트에 필요한 모든 폴더와 파일, 이미지, 비디오, 스프레드시트 및 데이터 세트가 포함될 수 있습니다. 리포지토리에 README 파일, 프로젝트에 대한 정보가 포함된 파일이 있는 경우도 많습니다. README 파일은 일반 텍스트 Markdown 언어로 작성됩니다 분기하면 한 번에 다른 버전의 리포지토리를 사용할 수 있습니다.

기본적으로 리포지토리에는 최종 분기로 간주되는 main이라는 하나의 분기가 있습니다. 리포지토리의 main에서 추가 분기를 만들 수 있습니다. 분기를 사용하여 한 번에 다른 버전의 프로젝트를 사용할 수 있습니다. 이 기능은 코드의 기본 소스를 변경하지 않고 프로젝트에 새 기능을 추가하려는 경우에 유용합니다. 다른 분기에서 수행된 작업은 병합할 때까지 주 분기에 표시되지 않습니다. 이 내용은 이 가이드의 뒷부분에서 설명합니다. 분기를 사용하여 분기를 main으로 커밋하기 전에 실험하고 편집할 수 있습니다. 원격 URL은 Git에서 “코드가 저장된 위치”를 나타내는 효율적인 방법입니다. 해당 URL은 GitHub, 다른 사용자의 포크 또는 완전히 다른 서버의 리포지토리일 수 있습니다.

다음 두 가지 유형의 URL 주소로만 푸시할 수 있습니다.

* HTTPS URL(예: https://github.com/user/repo.git)
* SSH URL(예: git@github.com:user/repo.git)

Git은 원격 URL을 이름과 연결하며, 기본 원격을 일반적으로 origin이라고 합니다. 분기에서 main 분기를 만들 때 해당 시점의 main 복사본 또는 스냅샷을 만듭니다. 분기에서 작업하는 동안 다른 사람이 main 분기를 변경한 경우 해당 업데이트를 가져올 수 있습니다. 이전 단계에서 새 분기를 만들면 GitHub에서 main의 사본인 새 readme-edits 분기의 코드 페이지로 이동합니다. 리포지토리에서 파일을 변경하고 저장할 수 있습니다. GitHub에서 저장된 변경 내용을 커밋이라고 합니다. 각 커밋에는 특정 내용을 변경한 이유에 대한 설명인 관련 커밋 메시지가 있습니다. 커밋 메시지는 다른 기여자가 사용자가 수행한 작업과 이유를 이해할 수 있도록 변경 내용의 기록을 캡처합니다. 변경 내용은 readme-edits 분기의 README 파일에만 적용되므로 이제 이 분기에는 main과 다른 콘텐츠가 포함됩니다. main의 분기가 변경되었으므로 끌어오기 요청을 열 수 있습니다.

끌어오기 요청은 GitHub 협업의 핵심입니다. 끌어오기 요청을 열면 변경 내용을 제안하고 누군가가 기여를 검토해서 끌어온 다음 해당 분기에 병합하도록 요청합니다. 끌어오기 요청은 두 분기 콘텐츠의 diff 또는 차이점을 표시합니다. 변경 내용, 더하기 및 빼기 항목은 서로 다른 색으로 표시됩니다. 커밋하는 즉시 코드를 완료하기 전에 끌어오기 요청을 열고 토론을 시작할 수 있습니다. 끌어오기 요청 메시지에서 GitHub의 @mention 기능을 사용하면 특정한 사람이나 팀이 복도 끝에 있든, 10시간의 시차가 있는 곳에 있든 상관없이 이들로부터 피드백을 요청할 수 있습니다. 사용자 고유의 리포지토리에서 끌어오기 요청을 열고 직접 병합할 수도 있습니다. 대규모 프로젝트에서 작업하기 전에 GitHub 흐름을 학습하기 좋은 방법입니다.

이 마지막 단계에서는 readme-edits 분기를 main 분기에 병합합니다. 끌어오기 요청을 병합하면 readme-edits 분기 관련 변경 사항이 main에 통합됩니다. 경우에 따라 끌어오기 요청은 main의 기존 코드로 충돌하는 코드를 변경할 수 있습니다. 충돌이 있는 경우 GitHub는 충돌하는 코드에 대해 경고하고 충돌이 해결될 때까지 병합을 방지합니다. 충돌을 해결하는 커밋을 만들거나 끌어오기 요청에서 주석을 사용하여 팀 구성원과 충돌을 논의할 수 있습니다. 이 연습에서는 충돌이 없으므로 분기를 주 분기에 병합할 준비가 된 것입니다. 명령줄에서 Git을 사용하려면 컴퓨터에 Git을 다운로드, 설치 및 구성해야 합니다. 명령줄에서 GitHub CLI를 설치하여 GitHub를 사용할 수도 있습니다.  Git에서 GitHub 리포지토리에 연결하는 경우 HTTPS 또는 SSH를 사용하여 GitHub으로 인증해야 합니다. HTTPS를 사용하여 복제하는 경우 자격 증명 도우미를 사용하여 Git에서 GitHub 자격 증명을 캐시할 수 있습니다. 현재 HTTPS를 사용하여 복제하는 것을 권장하고 있습니다. 연결을 하고 나면 이제 프로젝트를 사용할 리포지토리를 만들도록 선택할 수 있습니다. 리포지토리에 코드를 저장하면 코드를 백업하고 전 세계에 공유할 수 있습니다.