Git 기본 사용법

1. 저장소 생성 및 관리
   1. GitHub에서 프로젝트를 시작하려면 먼저 저장소를 만들거나 기존 저장소를 가져와야 합니다. 예를 들어, 저장소를 복사(Fork)하면 원본 프로젝트가 계정으로 복제됩니다. 그런 다음 로컬 컴퓨터로 저장소를 다운로드하여 작업할 준비를 할 수 있습니다.
   2. 원본 저장소와 연결하여 최신 변경 사항을 가져오는 방식도 중요합니다. 원본 저장소를 "upstream"이라고 부르며, 이를 설정하면 프로젝트의 동기화를 더 쉽게 관리할 수 있습니다.
2. 커밋
   1. Git에서는 파일을 수정한 다음 변경 내용을 저장합니다. 이 작업은 "커밋"이라고 불리며, 프로젝트의 수정된 파일에 대한 설명을 기록합니다. 각 커밋은 프로젝트의 히스토리를 기록하므로 협업할 때 변경 사항을 쉽게 추적할 수 있습니다.
3. 푸시
   1. 작업한 변경 사항은 로컬 환경에서만 저장되는데, 이를 공유하려면 원격 저장소로 보내야 합니다. 이를 "푸시"라고 합니다. 이렇게 하면 팀원들이 작업 내용을 볼 수 있습니다.
4. 풀
   1. 원격 저장소에 새로운 변경 사항이 생기면 이를 로컬 환경으로 가져와 작업을 계속할 수 있습니다. 이 작업을 "풀"이라고 하며, 팀 프로젝트에서 협업 시 자주 사용됩니다.
5. 브랜치 관리
   1. 브랜치란 프로젝트의 독립적인 작업 공간을 의미합니다. 새로운 브랜치를 만들어 실험적으로 작업할 수 있으며, 작업이 완료되면 프로젝트의 메인 브랜치로 병합할 수 있습니다. 필요 없는 브랜치는 삭제하여 저장소를 정리할 수 있습니다.
6. 충돌 해결 방법
   1. 프로젝트의 여러 사람이 동시에 작업하면 충돌이 발생할 수 있습니다. 이 경우 문제가 생긴 파일을 열어 적절히 수정해야 합니다. Git은 충돌 부분을 표시해주므로 이를 보고 변경 사항을 합의된 내용으로 수정합니다. 수정이 끝난 뒤, 저장하여 다른 팀원과 공유합니다.

브랜치 네이밍 규칙

* feat : 새로운 기능 추가, 기존의 기능을 요구 사항에 맞추어 수정
* fix : 기능에 대한 버그 수정
* build : 빌드 관련 수정
* chore : 패키지 매니저 수정, 그 외 기타 수정 ex) .gitignore
* ci : CI 관련 설정 수정
* docs : 문서(주석) 수정
* style : 코드 스타일, 포맷팅에 대한 수정
* refactor : 기능의 변화가 아닌 코드 리팩터링 ex) 변수 이름 변경
* test : 테스트 코드 추가/수정
* release : 버전 릴리즈

커밋 메시지도 동일하게 써 주시면 됩니다.

협업 가이드 라인

1. 코드 작성 및 커밋 메시지
   1. 간결하고 작업 내용을 설명하도록 작성
   2. 예시 ) [feat]인벤토리 기능 추가
2. Pull Request 프로세스
   1. 작업 완료 후 원격 저장소에 업로드 하고 Pull Request를 생성함
   2. Pull Request에는 작업내용, 변경사항 및 목적을 명확히 작성
   3. 팀원들과의 코드 리뷰를 통해 품질을 확인 후 병합
3. 정기적인 업데이트
   1. 작업 시작 전에 항상 최신 변경사항을 로컬 브랜치로 가져옴
   2. 팀원들과 주기적으로 진행 상황을 공유함.

예외 상황 처리법

1. 충돌 발생
   1. 충돌이 발생하면 Git은 자동으로 충돌 파일을 표시함.
   2. 충돌 파일을 열어 표시된 내용을 검토한 후 수정함.
   3. 해결 한 뒤 커밋 하고 팀원과 공유하여 확인 받음.
2. 긴급 수정
   1. 배포 후 발견된 긴급한 문제는 Hotfix 브랜치를 생성하여 처리
   2. 수정 작업을 완료하면 Main브랜치에 병합하여 모든 팀원에게 변경하상을 반영함
3. 작업 중단 또는 병합 문제
   1. 작업 중 심각한 문제가 발생하면 즉시 팀원들에게 상황을 알린다.
   2. 문제가 해결될 때 까지 관련 브랜치를 잠그거나 작업을 중단하고 해결 방법을 논의함
   3. 병합 문제는 코드 리뷰를 통해 함께 해결함.
4. 원격 저장소와의 동기화 문제
   1. 원격 저장소와 로컬 저장소의 상태가 일치하지 않는 경우, 최신 상태로 동기화를 시도하고 문제가 지속되면 팀원과 협력하여 해결함.