

GitHub란?

1. Github의 역사와 배경

GitHub는 2008년에 Tom Preston-Werner, Chris Wanstrath, PJ Hyett에 의해 창업되었다. GitHub의 목적은 개발자들이 소프트웨어 개발과 협업을 더욱 효과적으로 수행할 수 있는 플랫폼을 제공하는 것이다.

GitHub는 Git이라는 분산 버전 관리 시스템을 기반으로 하는 효과적인 소프트웨어 개발 및 협업 도구이다. Git은 분산 버전 관리 시스템으로, 소스 코드의 변경 이력을 추적하고 여러 개발자가 동시에 작업할 수 있는 기능을 제공한다.

GitHub의 성장은 많은 개발자들과 기업들이 GitHub를 사용하여 소스 코드를 저장하고 공유하며 협업하는데 활용하였다. 이러한 성과로 GitHub는 현재 오픈 소스 생태계에서 가장 중요하고 폭넓게 사용되는 플랫폼 중 하나가 되었다.

2018년에 GitHub는 Microsoft의 자회사로 들어가 이후에도 GitHub는 계속해서 발전하며 개발자 커뮤니티와 협력하여 소프트웨어 개발과 협업을 지원하고 있다.

2. GitHub의 주요기능과 이점

- **원격 저장소 기능:** 원격 저장소는 코드 및 프로젝트 파일을 인터넷상의 다른 위치에 저장하고 관리하는 기능이다. Git을 사용하는 경우는 원격저장소가 중앙 저장소 역할을 수행한다. GitHub는 Git을 기반으로 하여 소스 코드를 원격 저장소에 저장하고 관리할 수 있다. 개발자는 자신의 로컬 컴퓨터에서 작업한 코드를 원격저장소에 푸시하여 다른 개발자나 팀원들과 공유하고 동기화할 수 있다. 원격저장소는 로컬환경에서 데이터를 안전하게 보관하고 협업을 원활하게 진행하는데 다음과 같은 이점을 제공한다.

- ☞ **백업과 복원:** 원격 저장소는 데이터의 백업과 복원을 용이하게 하여 로컬 컴퓨터나 장치에 저장된 데이터가 손실되더라도 원격 저장소에 백업된 데이터를 사용하여 손실을 최소화하고 복구 할 수 있도록 도와준다.
- ☞ **원격 접근과 이동성:** 원격 저장소를 사용하면 어디서든 데이터 접근할 수 있다. 인터넷에 연결된 어떤 장치나 컴퓨터에서도 데이터를 확인하고 수정할 수 있으므로 팀원들은 자유롭게 작업을 진행하고 협업할 수 있다.
- ☞ **보안과 안정성:** 원격 저장소는 보안 및 데이터의 안정성을 제공한다. 일반적으로 원격 저장소는 데이터의 암호화, 접근 제어, 백업 등의 보안 기능을 제공하여 데이터의 손실이나 무단 액세스로부터 보호한다. 또한, 여러 사용자가 함께 작업하면서 발생하는 문제를 방지하기 위해 안전성을 유지할 수 있다.

- **버전 관리 기능:** 버전 관리는 소프트웨어 개발 및 파일 관리에서 변경사항을 추적하고 관리하는 기능이다. 버전 관리 시스템은 코드, 문서, 그래픽 디자인 파일 등 모든 종류의 파일을 추적하여 파일의 변경 내역을 기록하고, 필요한 경우 이전 버전으로 롤백하거나 변경사항을 비교하고 병합하는 등의 작업을 지원한다. 버전 관리 기능의 주요 이점은 다음과 같다.

- ☞ **변경 이력 추적:** 버전 관리 시스템을 사용하면 모든 파일과 코드의 변경 이력을 추적할 수 있다. 각 변경사항은 세밀한 단위로 기록되며, 누가, 언제, 어떤 변경을 수행했는지를 알 수 있다. 이를 통해 개발자는 이전상태로의 롤백이나 특정 시점의 코드를 비교할 수 있다.
- ☞ **협업과 공유:** 버전 관리 시스템을 사용하면 여러 개발자가 동시에 작업하고 변경 사항을 공유할 수 있다. 각 개발자는 자신이 작업한 내용을 커밋(commit)하여 다른 팀원들과 쉽게 공유하고 통합할 수 있습니다. 변경사항에 대한 명확한 커밋 메시지와 주석을 통해 코드의 의도를 파악하고 이해하기 쉽게 도와준다. 이를 통해 팀원 간의 협업이 원활해지고, 코드의 충돌이나 중복 작업을 방지할 수 있다.
- ☞ **버전 관리와 분기:** 버전 관리 시스템을 사용하면 개발 프로젝트의 다양한 버전을 관리할 수 있다. 주요기능 개발, 버그 수정, 안정화 등의 목적에 따라 프로젝트를 분기하여 각각의 버전을 개별적으로 관리할 수 있습니다. 이를 통해 특정 버전의 코드를 유지 보수하거나 이전 버전으로 롤백할 수 있다.
- ☞ **안정성과 신뢰성:** 버전 관리 시스템은 변경사항을 안전하게 보관하고 관리하여 소프트웨어의 안정성과 신뢰성을 높인다. 이전 버전으로의 롤백이나 변경사항의 추적을 통해 문제가 발생한 경우 원인 분석과 복구 작업을 쉽게 할 수 있다.

- **프로젝트 관리기능:** 프로젝트 관리기능은 프로젝트 계획, 추적, 조정하는데 도움을 주는 다양한 도구와 기능을 포함한다. 프로젝트 관리기능은 프로젝트를 효율적으로 관리하고 조정하는데 많은 이점을 제공한다.

- ☞ **일정 및 마감일 관리:** 프로젝트 관리기능은 작업과 활동을 일정에 기록하고 추적하여 프로젝트의 일정을 계획하고 조정하는데 도움을 주어 프로젝트 팀은 작업의 우선순위와 마감일을 알고, 작업을 조기에 식별하여 시간을 효율적으로 관리할 수 있도록 한다.
- ☞ **작업 할당 및 협업:** 프로젝트 관리기능은 프로젝트에 필요한 자원을 관리하고 할당하여 자원의 효율적인 사용과 예산조절을 가능하게 한다. 또한, 자원의 가용성과 활용도를 파악하여 프로젝트 성공 가능성을 높인다.
- ☞ **문서관리:** 프로젝트와 관련된 문서를 중앙에서 관리하고 공유하여 팀원들은 필요한 정보에 쉽게 접근하고, 업무에 필요한 문서를 신속하게 공유하며 문서의 버전 관리와 협업을 원활하게 한다.
- ☞ **리스크 관리:** 프로젝트 관리기능은 프로젝트 진행 중 발생할 수 있는 리스크를 식별하고 관리하여 프로젝트 팀이 예측할 수 없는 문제에 대비하고, 필요한 조치를 준비할 수 있도록 하여 프로젝트의 성공확률을 높여준다.

- **이슈 트래킹 기능:** 이슈트래킹은 프로젝트의 문제 해결과 개선과정을 체계적으로 추적하고 관리하기 위한 중요한 도구이다. GitHub는 이슈 트래킹 시스템을 제공하여 버그 리포트, 개선 제안, 작업 항목 등을 관리할 수 있도록 한다. 개발자가 문제점이나 개선사항들(이슈)을 생성하고 할당하며, 댓글과 레이블 등을 사용하여 상세 정보와 상태를 확인할 수 있다. 이슈 트래킹 기능은 프로젝트나 작업에서 발생하는 이슈를 추적하고 관리하는 데 많은 이점을 제공한다.

- ☞ **문제 식별 및 추적:** 이슈 트래킹을 통해 프로젝트에서 발생하는 문제를 식별하고 추적할 수 있다. 이슈를 등록하고 상태를 업데이트 함으로써 팀원들은 어떤 문제가 발생하고 있는지 파악하고 이를 관리하며 해결책을 찾을 수 있다.
- ☞ **우선순위와 할당:** 이슈 트래킹은 문제에 대한 우선순위를 설정하고 책임을 할당하여 어떤 이슈에 집중해야 하는지 알 수 있고 작업을 분배하고 추적하여 문제해결을 원활하게 할 수 있도록 도와준다.
- ☞ **협업과 투명성:** 이슈 트래킹 시스템은 팀원들간의 협업과 의사소통을 촉진하여 이슈에 대한 의견이나 피드백을 공유하도록 하고 해당 이슈에 대한 업데이트를 확인할 수 있도록 한다.
- ☞ **문제 예방과 개선:** 이슈 트래킹은 반복적으로 발생하는 문제나 패턴을 파악하는 기능이 있어 팀이 문제 예방을 위한 조치를 취할 수 있도록 하고 프로세스나 제품의 개선을 위한 정보를 얻을 수 있도록 한다.

- **코드 리뷰 기능:** 코드 리뷰기능은 소프트웨어 개발과정에서 작성된 코드를 동료 개발자나 팀원들이 검토하고 평가하는 기능이다. GitHub는 코드 리뷰 기능을 통해 개발자들이 코드를 검토하고 의견을 주고받을 수 있는 환경을 제공한다. 개발자들은 변경사항을 비교하고 주석을 달며 피드백을 제공하여 코드 품질을 향상시킬 수 있다. 주로 협업 환경에서 사용되며, 다음과같은 이점을 가지고 있다.

- ☞ **지식 공유:** 개발자들 간의 지식 공유를 촉진하여 다른 개발자들의 코드를 검토하고 이해하는 과정에서 새로운 개념, 패턴, 최적화 기법 등을 학습하고 팀전반에 지식을 확산시킬 수 있다.
- ☞ **품질 향상:** 코드 리뷰를 통해 개발자들은 작성한 코드의 품질을 개선할 수 있다. 다른 개발자들이 코드를 검토하고 피드백을 제공하므로 잠재적인 버그, 취약성, 오용 또는 코드 스타일 등의 문제를 발견하고 수정할 수 있다.
- ☞ **일관성 유지:** 코드 리뷰 기능은 팀 내에서 일관성있는 코딩 스타일과 가이드라인을 준수할 수 있도록 도와준다. 개발자들은 리뷰어로부터 피드백을 통해 일관성있는 코딩 규칙을 준수하고 팀의 코드 베이스를 통일된 상태로 유지할 수 있다.
- ☞ **버그 예방:** 코드 리뷰를 통해 개발자들은 잠재적인 버그를 조기에 발견하고 수정할 수 있다. 리뷰어들은 코드를 신중하게 분석하고 테스트환경을 기반으로 문제를 찾을 수 있으며, 이것은 실제 환경에서의 버그 발생을 방지하는데 도움을 준다.

- **협업 기능:** 협업 기능은 여러 개발자가 함께 작업하고 정보를 공유하여 효율적으로 협업하는 데에 도움을 주는 기능들의 집합이다. 소프트웨어 개발 프로젝트에서 협업 기능은 팀원들 간의 커뮤니케이션, 작업 조율, 코드 리뷰, 작업추적 등을 지원하는 다양한 도구와 기능을 포함한다. GitHub에서도 다수의 개발자들이 동시에 작업하고 협업할 수 있는 기능을 제공한다. 개발자들은 브랜치(branch)를 생성하여 독립적으로 작업한 후 변경사항을 병합(merge)하여 하나의 코드베이스로 통합할 수 있다. 이러한 협업기능은 개발 프로젝트의 투명성과 생산성을 향상시키는 데에 중요한 역할을 한다. 협업 기능의 주요 이점은 다음과 같다.

- ☞ **효율적인 업무 분담:** 협업 기능을 통해 팀원들은 작업을 효율적으로 분담할 수 있다. 작업을 각자 나누어 맡아 동시에 진행하거나, 우선순위에 따라 작업을 할당하고 추적할 수 있도록하여 작업량을 분산시키고 업무를 더욱 효과적으로 처리할 수 있도록 한다.
- ☞ **실시간 커뮤니케이션:** 협업 기능은 실시간 커뮤니케이션을 지원한다. 팀원들은 채팅, 댓글, 토론 포럼 등을 통해 의견을 주고받고, 질문을 해결하며, 의사 결정을 내릴 수 있다. 이를 통해 의사소통을 원활하게 하고 의견조율을 할 수 있다.
- ☞ **동시 작업 및 충돌 방지:** 협업 기능은 동시에 여러 팀원이 작업을 할 수 있도록 지원한다. 여러 개발자가 동시에 동일한 파일이나 프로젝트를 수정해도 충돌이 발생하지 않도록 관리하고, 변경사항을 통합하여 충돌을 방지할 수 있도록 한다.
- ☞ **역할과 권한 관리:** 협업 기능은 역할과 권한 관리를 제공한다. 프로젝트에 따라 팀원들에게 적절한 역할과 권한을 할당하여 접근 제어를 설정할 수 있도록하여 보안을 강화하고 프로젝트 관리를 더욱 효과적으로 수행할 수 있도록 도와준다.

- **웹 호스팅 기능:** 웹 호스팅 기능은 웹 사이트나 웹 애플리케이션을 인터넷 상에서 접근 가능하게 하는 기능을 말한다. GitHub는 정적 웹 사이트 호스팅 기능을 제공한다. 개발자들은 GitHub에 HTML, CSS, JavaScript 등의 웹 페이지 파일을 업로드하여 간단한 웹 사이트를 호스팅할 수 있다. 웹 호스팅 기능은 다음과 같은 기능과 이점을 가진다.

- ☞ **서버관리:** 웹 호스팅은 사용자가 웹 브라우저를 통해 웹 사이트에 접속할 수 있도록 서버 공간을 제공하고 관리하여 서버의 안정성과 신뢰성을 유지할 수 있는 장점을 제공한다.
- ☞ **데이터 백업:** 웹 호스팅은 주기적으로 데이터를 백업하여 웹 사이트의 데이터 손실을 방지하여 잠재적인 문제나 장애 상황에서도 데이터를 복구할 수 있는 안전성을 제공한다.

3. GitHub의 역할과 중요성

GitHub는 개발자들이 소스코드를 공유하고 협업하는 중심지이다. 개발자들은 원격저장소를 통해 프로젝트를 협업함으로써 지식 공유를 가능케 한다. GitHub는 개방적인 생태계를 가지고 있어 수많은 오픈 소스 프로젝트들이 GitHub를 통해 공개되고 관리된다. 개발자들은 다양한 프로젝트에 참여하고 기여하며, 커뮤니티와의 교류를 통해 학습과 지식 공유의 기회를 얻을 수 있다. GitHub의 역할과 중요성은 소프트웨어 개발 생태계에 큰 영향을 미치고 있다. 협업과 형상 관리, 이슈 추적, 코드리뷰, 오픈소스 기여 등 다양한 측면에서 개발자들에게 필수적인 플랫폼으로 자리잡고 있다.