

# Git Flow

branch	주요 특징	예시
Master (main)	<ul style="list-style-type: none"><li>배포 가능한 상태의 코드</li></ul>	<b>LOL</b> 클라이언트 라이브 버전 <b>(9.23.298.3143)</b>
Develop (main)	<ul style="list-style-type: none"><li><b>feature branch</b>로 나뉘어지거나, 발생한 버그 수정 등 개발 진행</li><li>개발 이후 <b>release branch</b>로 갈라짐.</li></ul>	다음 패치를 위한 개발 <b>(9.24)</b>
feature branches (supporting)	<ul style="list-style-type: none"><li>기능별 개발 브랜치 <b>(topic branch)</b></li><li>기능이 반영되거나 드랍되는 경우 브랜치 삭제</li></ul>	개발시 기능별 예) 신규챔피언 세나, 드래곤 업데이트
release branches (supporting)	<ul style="list-style-type: none"><li>개발 완료 이후 <b>QA/Test</b> 등을 통해 얻어진 다음 배포 전 <b>minor bug fix</b> 등 반영</li></ul>	<b>9.24a, 9.24b, ...</b>
Hotfixes (supporting)	<ul style="list-style-type: none"><li>긴급하게 반영 해야하는 <b>bug fix</b></li><li><b>release branch</b>는 다음 버전을 위한 것이라면, <b>hotfix branch</b>는 현재 버전을 위한 것</li></ul>	긴급 패치를 위한 작업 예) 버그로 인한 챔피언 선택 금지

Happy Hacking

# Git Flow

**Git Flow는 정해진 답이 있는 것은 아니다.**

**Github Flow, Gitlab Flow 등의 각 서비스별 제안되는 흐름이 있으며,**

**변형되어 각자의 프로젝트/회사에서 활용 되고 있다.**

**간단하게 브랜치를 활용하는 명령어를 알아보고,**

**프로젝트에 활용할 수 있는 간단한 버전의 브랜치 전략을 배워보자.**