

# 팀의 브랜치 전략

- 메인브랜치: Master
  - 배포 혹은 빌드되어 출시 가능한 버전의 코드만 업로드
- 서브 브랜치: Develop
  - Develop 브랜치는 크게 기능별로 실시간 정보 조회 & 활용 & 시각화로 나눈다.
  - 세부 기능은 .py를 나눌 예정
    - 예시) 실시간 정보 조회.py, 실시간 정보 저장.py, 정보 정제.py, 정보 활용.py
    - 각각의 .py는 import로 연결 ( 예시. `from tool import csvWriter` )
  - Develop는 Master의 하위 브랜치
    - 개발한 .py를 Develop의 "commit-pull request"를 한 후, 리뷰를 달아서 검수
    - 최종적으로 Master에서 MERGE
- 우선 순위: 기능별 브랜치를 따는 방식 > 개발자별 브랜치를 따는 방식

# 현재까지 리서치해봤거나 수행해본 내용

- 깃허브 세팅

- Manage access 권한 승인
  - 모든 팀원이 github 커밋 가능
- VS code 에서 github 환경 세팅 (or Github Desktop)
  - Git commit & pull: 테스트용으로 "AIFT\_팀플첫회의.pdf" (작성파일) 파일 업로드
- .gitignore: python
- .readme => .py의 파일 위치 및 기능 설명할 예정

- 키움 API 조사 from Github

- <https://wikidocs.net/84131>
- TR 연속 조회
  - <https://github.com/sharebook-kr/pykiwoom>
- 실시간 TR 조회
  - <https://github.com/elbakramer/koapy>