







# 2023-하계방학 자기주도학습동아리 활동보고서

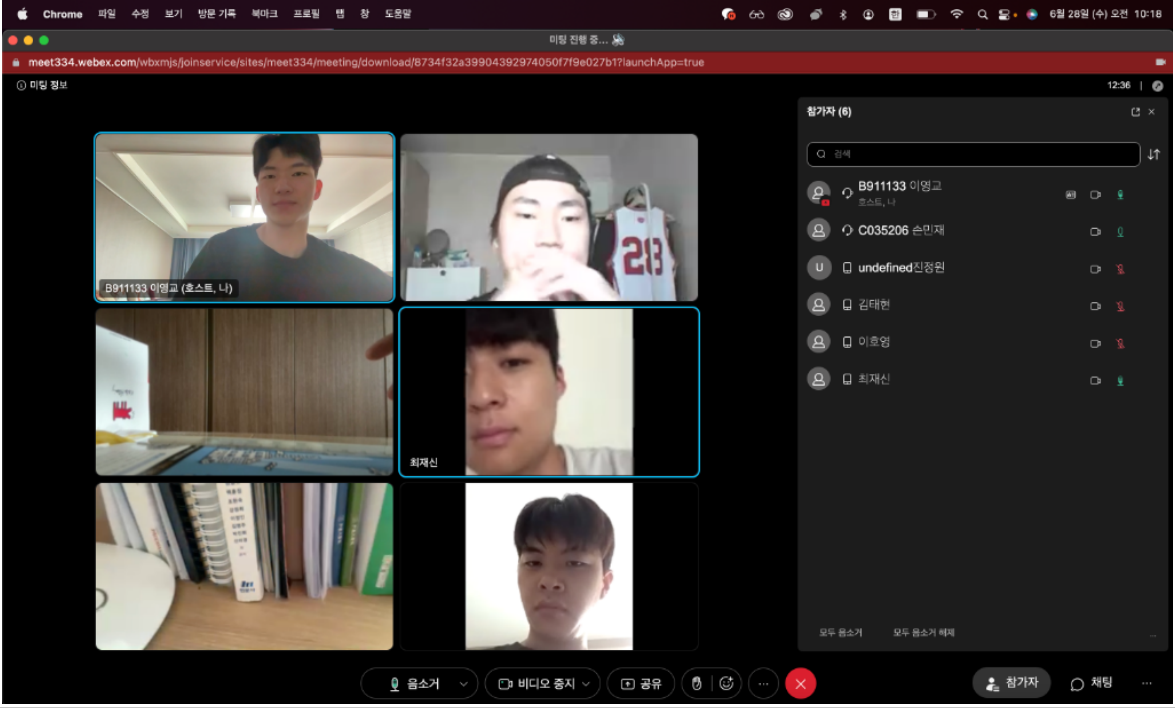
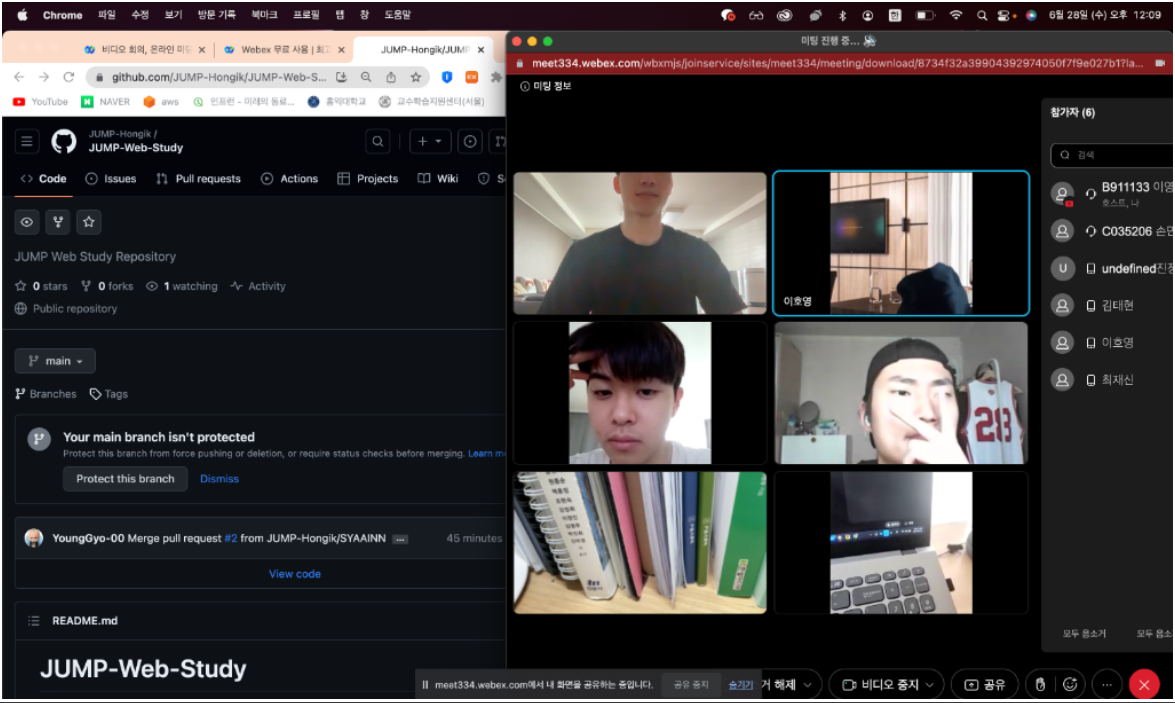
팀명	JUMP		
학습일시	2023년 06월 28일 (목요일)	진행차수	1주차
학습시간	10시 00분 ~ 12시 00분 (2시간)		
참석자	이 영 교 	김 태 현 	손 민 재 
	이 호 영 	최 재 신 	진 정 원 
불참자	X		
구 분	내 용		
학습목표	<ul style="list-style-type: none"> <li>- 혼자 개발하는 것이 아닌, 여러 사람들과 협업하는 과정에서 Git을 통해 코드를 공유할 수 있다.</li> <li>- 코드를 공유하는 과정 속에서, Commit Message 만을 보고도 해당 Commit은 어떤 코드를 변경한 것인지 쉽게 파악할 수 있게 Git Commit Message Style Guide를 정해보자.</li> </ul>		
학습 방법 및 학습 내용	<p>1. 학습 방법 : '지옥에서 온 Git' 학습 사이트를 바탕으로 개인적으로 일주일동안 협업할 때 사용 가능한 Git에 대해 공부한 후, 다른 조원들에게 알려주고 싶은 내용들을 정리하여 Notion Team Space에 정리해서 올린다. 추가로, Github Organization에 공부한 코드를 공유하며 일주일동안의 스터디 내용을 공유한다. 마지막으로, 디스코드에서 회의를 통해 2시간 동안 자신이 공부한 분량을 소개한다.</p> <p>** Git 학습 사이트 : <a href="https://opentutorials.org/course/2708">https://opentutorials.org/course/2708</a>  ** 1주차 회의록 노션 링크 : <a href="https://citrine-situation-94c.notion.site/1-2023-06-28-4aee65b9be5244e28327940935c8e508?pvs=4">https://citrine-situation-94c.notion.site/1-2023-06-28-4aee65b9be5244e28327940935c8e508?pvs=4</a>  ** Github 링크 : <a href="https://github.com/JUMP-Hongik">https://github.com/JUMP-Hongik</a></p> <p>2. 학습 내용 : 1) Git을 사용하여 많은 사람들과 코드를 공유할 때 Commit Message의 구조를 정해서 동일한 구조로 작성해야 한다. 여러 사람들이 Commit Message 만을 보고도 해당 Commit은 어떤 코드를 변경하였는지 이해할 수 있어야 하기 때문이다. 따라서, 우리 팀은 Commit Message Structure를 Type : subject, body, footer로 3가지 단락으로 구분하여 수정한 코드의 역할을 분명히 전달하기로 했다.</p> <h3>1. Commit Message Structure</h3> <div style="background-color: #f0f0f0; padding: 10px; margin: 10px 0;"> <pre> type : subject  body  footer </pre> </div> <p>2) Commit Type은 자신이 코드를 수정한 것이 어떤 역할인지 구분하는 키워드를 나타내는 것으로, 예시를 들어 설명하면 Feat은 새로운 기능을 추가하였다는 의미, Fix는 코드의 버그를 수정했다는 것을 의미하는 등 키워드에 각각 담긴 의미들을 설정하고 맨 앞부분에 붙임으로써, Commit이 어떤 것을 의미하는지 요약해주는 역할을 한다.</p>		



	<ul style="list-style-type: none"> <li>• <b>Feat</b> : 새로운 기능 추가</li> <li>• <b>Fix</b> : 버그 수정</li> <li>• <b>Docs</b> : 문서 수정</li> <li>• <b>Style</b> : 코드 포매팅, 세미콜론 누락</li> <li>• <b>Refactor</b> : 코드 리팩토링</li> <li>• <b>Test</b> : 테스트 코드</li> <li>• <b>Chore</b> : 빌드 업무 수정</li> </ul> <p>3) Subject(제목)은 최대 50글자가 넘지 않도록 하고, 마침표 및 특수 기호는 사용하지 않도록 유의한다. 영문으로 표기하는 경우 동사(원형)를 가장 앞에 두고, 첫 글자는 대문자(과거 시제 X) 제목은 개조식 구문으로 작성 -&gt; 서술형 문장은 쓰지 말기로 한다. 이는 간결하고 요점적인 서술을 위해서 이렇게 만들었다.</p> <p>4) Body는 내용의 양에 구애받지 않고 최대한 상세히 작성하도록 한다. 왜 변경했고, 어떻게 변경했는지 등을 간단하게 설명하는 식으로 작성하도록 하자.</p> <p>5) 마지막 Footer는 적지 않아도 되며, 참고할 이슈 사항이나 이슈 해결 등을 적어주는 역할을 한다.</p> <p>팀원 전체가 이 컨벤션을 지켜가며 코드를 공유하는 습관을 익히고 시작했으면 좋겠다는 생각에 컨벤션을 정하는 시간을 1주차에 가져갔다.</p> <p>추가로, 각자 사이트를 통해 배운 내용들에 대해 이해가 되지 않는 부분들을 회의하는 시간 동안에 다른 팀원들에게 실습을 하는 과정을 보여주며 이해하는 시간을 가졌다. 앞으로 코드를 공유할 때 자신만의 Branch 를 개설하여 그 속에 코드를 올리고, 1주씩 회의 전 Branch를 병합하는 Pull Request 를 보내면, 팀장이 코드에 문제가 있는지 확인 후 병합하는 과정으로 진행하기로 약속했다.</p>
--	--

<b>활동 후기</b> * 본인이 맡은 역할 활동과 학습한 내용에 대한 후기 작성	팀장 / 팀원 이름	작성란
	이영교	팀장으로, 다른 팀원들의 회의 내용을 기록하며 해당 주차에 준비해온 프론트엔드 개발 지식들을 공유하는 역할을 했습니다. 처음으로 하는 스터디로 틀을 정하는데 어려움을 조금 겪었지만, 다음 주차부터는 더 짧은 시간 내로 마무리 할 수 있을 것 같다.
	김태현	1주차는 GIT 사용법과 JAVA 기본 내용에 대하여 복습하였습니다. 이전까지 했던 공부와는 다르게 학우들과 협업을 통해 이루고자 하는 목표가 있어 더욱 책임감 있게 공부할 수 있었습니다.
	손민재	Git, Github 의 기능 및 특성을 공부하면서, 프로젝트 협업 개발의 환경을 조금이나마 체험해보았습니다. 프로그램의 사용이나 스터디 자체가 처음이라 낯설었지만 재밌었고, 적응을 위해 반복 숙달이 필요하겠다고 생각했습니다.
	이호영	웹 사이트의 디자인을 구상하며 UX가 어떤 것인지 처음 알게 되었습니다. 사용자가 웹 사이트에 들어와서 행동하는 과정을 상상해보며 플로우를 그리는 과정에서 고려해야 할 사항들이 정말 많고, 웹 디자인하는 기술도 어렵다는 걸 알게 되었습니다.
	최재신	다른 개발하는 팀원들의 지식을 공유받았는데, 매번 사용하던



		<p>Github 에 Commit 하는 메시지의 컨벤션이 있다는 것을 처음 알게 되었다. 확실히 아무렇게나 작성해서 올리는 것보다 체계적으로 작성하는 것이 깔끔하고 좋았다.</p> <p>처음 Git에 대해 공부해봤는데, 코드를 Github에 Push 하는 과정을 이해하기 어려웠다. 사이트에 있는 자세한 설명이 아니었으면 시작부터 힘들었을 것 같음.</p>
<div>활동 증빙 사진</div>		<div> <div>진정원</div> </div> <div> <div>학습시작</div>  </div> <div> <div>학습종료</div>  </div> <div> <div>학습활동</div> </div>



