

# 프로젝트 명세서

블록체인 개발 입문

목차

1. 프로젝트 개요..... 3

2. 과제 목표 ..... 4

3. 학습 자료 ..... 5

4. 과제 ..... 9

5. 과제 설명 ..... 10

6. 산출물 제출 ..... 14

# 1. 프로젝트 개요

---

## 소개

블록체인 개발은 오늘날 세계에서 가장 빠르게 성장하고 있는 기술 분야 중 하나입니다. 실제로 Gartner 에 따르면 2024 년 까지 대기업의 20%가 결제, 가치저장 또는 담보로 디지털 화폐를 사용할 것이라고 보고 있습니다.

본 프로젝트는 블록체인을 개발하기 위한 입문 과정으로 간단한 블록체인의 원리와 개발 환경을 구성하고, 실습으로 블록체인을 코딩하여 만들어 봅니다.

본 프로젝트는 두 가지 파트로 나눠 진행하게 됩니다.

첫 번째, 블록체인을 개발하기 위한 기초 학습과 개발 환경 구성을 진행합니다.

두 번째, 간단한 블록체인을 코딩하여 생성해 봅니다.

## 2. 과제 목표

---

블록체인이란 무엇인지, 어떻게 동작하는지 학습하고, 블록을 생성하여 결과를 제출해야 합니다.

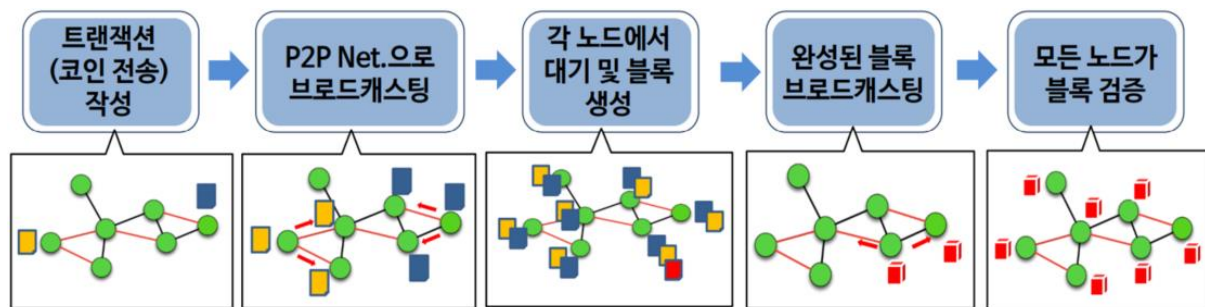
### 3. 학습 자료

블록체인이란 데이터 분산 처리 기술로 데이터를 암호화하여 저장하는 하나의 단위인 블록에 데이터를 담아 체인 형태로 연결하여 수많은 컴퓨터에 동시에 이를 복제해 저장하는 기술입니다.

누구라도 임의로 수정할 수 없고, 누구나 변경의 결과를 열람할 수 있습니다.

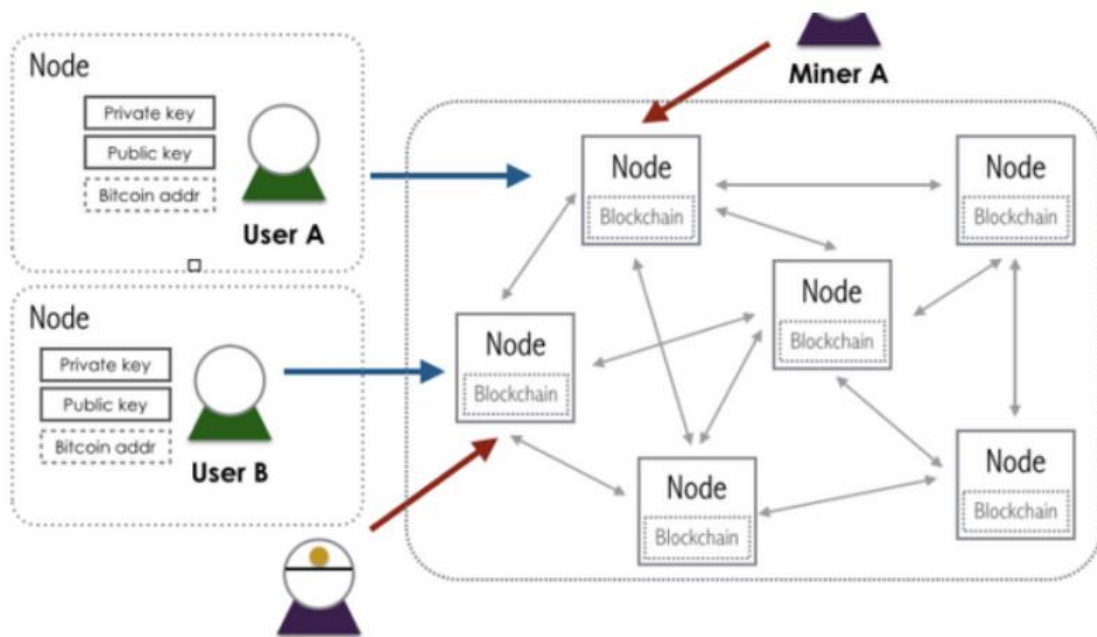
블록체인의 작동원리는 다수의 온라인 거래 기록을 묶어 하나의 데이터 블록(Block)을 구성한 뒤 해시(Hash) 값을 이용하여 이전 블록과 체인(Chain)처럼 연결합니다.

좀 더 자세히 알아보기 위해 아래 그림을 참고하여 설명하겠습니다.



하나의 블록은 헤더와 거래(트랜잭션) 리스트로 이루어져 있습니다. 각 거래기록의 최소 단위를 트랜잭션이라고 하며 트랜잭션이 만들어 지면 P2P로 모든 노드에 브로드 캐스팅을 합니다.

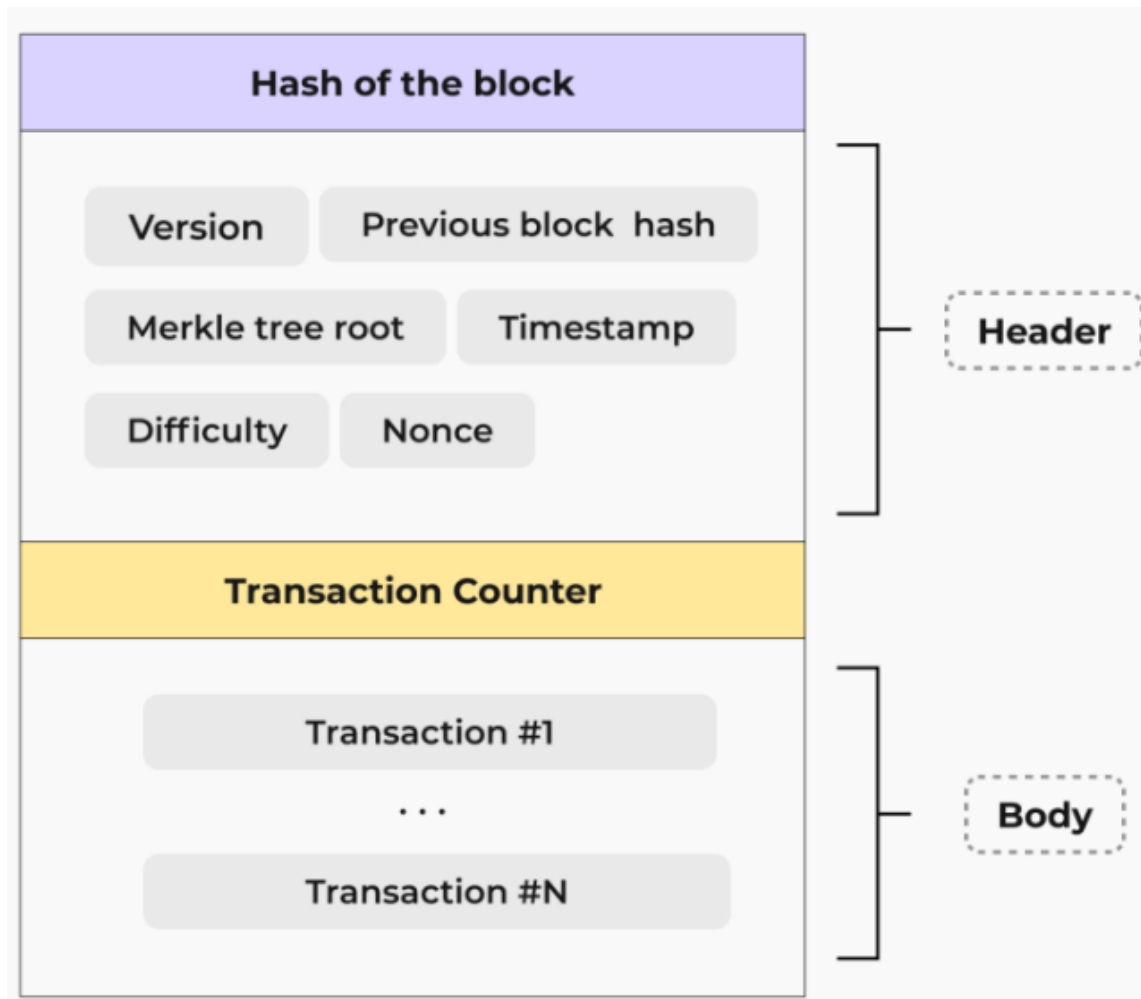
이때 노드는 블록체인 네트워크에 연결되어 있는 모든 서버를 의미하며, 각 노드에서는 거래기록을 모아 블록을 생성하게 됩니다.



각 노드에서 만든 블록이 모두 허용되는 것은 아니며, 주어진 난이도에 따라 블록의 해쉬값을 계산하여 조건에 가장 부합하는 블록이 새롭게 체인에 추가되는 블록으로 선택되게 되며 이러한 과정을 채굴 또는 마이닝이라고 합니다.

이렇듯 무작위로 선정된 검증인을 기반으로 하는 블록체인 네트워크에 대하여 새로운 블록을 추가하는 합의 메커니즘(알고리즘)을 작업 증명(PoW)이라고 합니다.

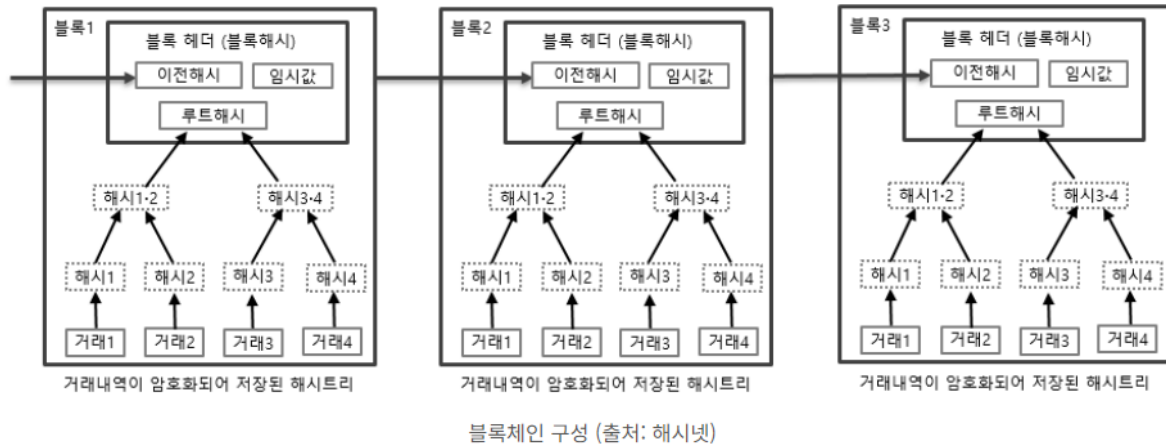
이제 블록이 어떻게 구성되어 있는지 살펴 보겠습니다.



각 블록은 Header 와 Body 로 구성되어 있으며, 블록의 대부분은 트랜잭션으로 구성되어 있습니다.

Header 를 구성하여 있는 이전 블록 해시 값, 머클트리 루트, 난이도, nonce(Nonce) 값이 무엇인지 학습을 진행해 주십시오. (과제 1)

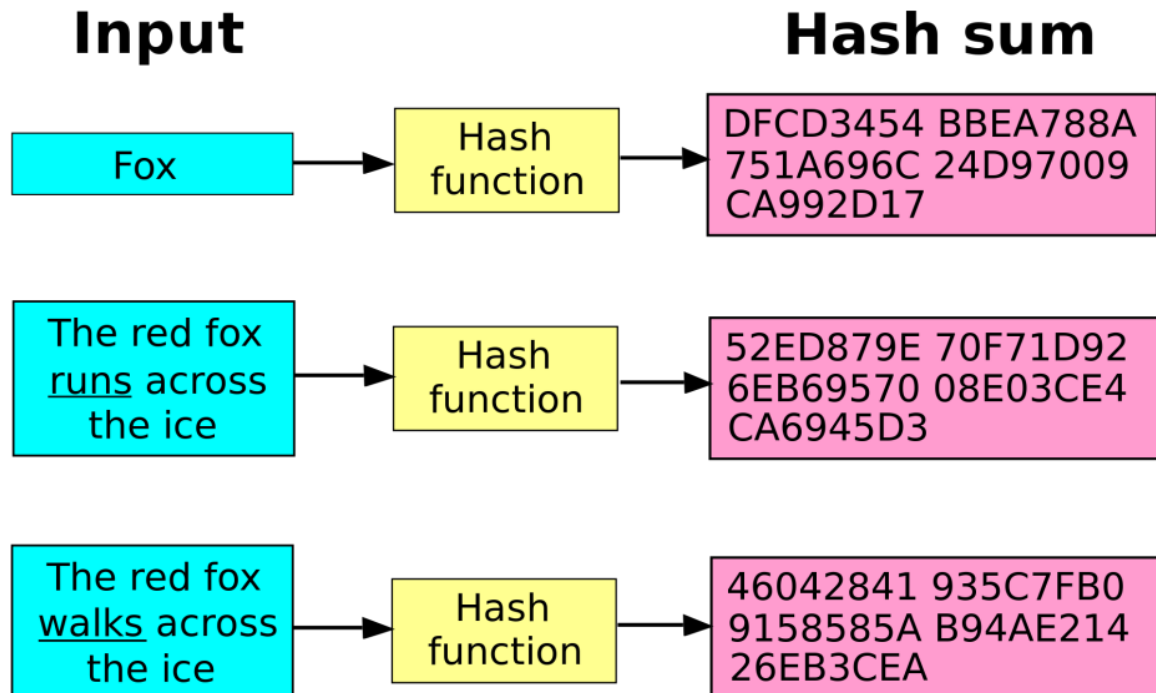
블록체인 구성은 아래와 같습니다.



마지막으로 블록 해시에 대해서 알아보겠습니다.

블록 해시는 블록 헤더 정보를 입력 값으로 SHA256 해시 함수를 적용해서 계산되는 값으로 해시 값은 해시 함수를 사용하여 32 바이트의 숫자 값을 출력합니다.

블록체인에서는 블록 전체를 해시 한 값이 아니라 블록 헤더를 해시 한 값을 사용합니다.





## 4. 과제

---

본 프로젝트에서는 두 가지 과제를 제출해야 합니다.

1. 챗터 3 에서 표시했던 이전 블록 해시 값, 머클트리 루트, 난이도, 논스(Nonce) 값이 무엇인지 학습을 진행하고 각각 정리된 내용을 제출 해 주십시오.
2. 두 번째 과제는 이더리움 테스트 네트워크 중 개인 테스트용 사설 테스트 넷을 구성하고 화면을 캡처 하여 제출 해 주십시오.
3. 2 번 과제를 수행하면서 만든 제네시스 블록 파일을 제출 해 주십시오.

## 5. 과제 설명

---

사설 네트워크 구축하기 위하여 제네시스 블록 파일과 블록 데이터 폴더가 필요합니다.

블록체인에서 가장 먼저 생성되는 블록을 제네시스 블록이라고 합니다.

이더리움 오픈소스 go-ethereum 는 제네시스 블록 파일은 JSON 형식의 파일로 만들게 됩니다.

⇒ <https://github.com/ethereum/go-ethereum> README.md 참조

⇒ <https://github.com/zaeem4/private-eth-1.0> README.md 참조

그럼 이제부터 테스트용 사설 테스트 네트워크를 구성해 보겠습니다.

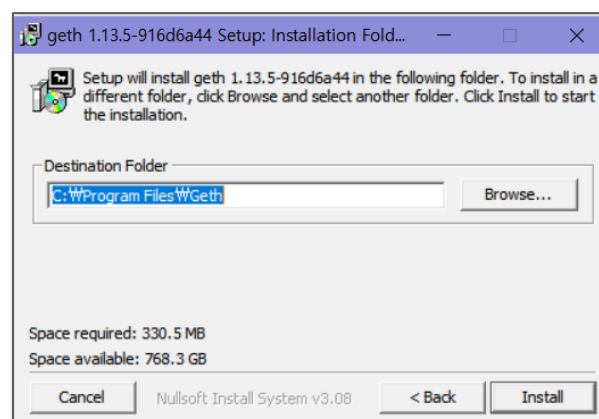
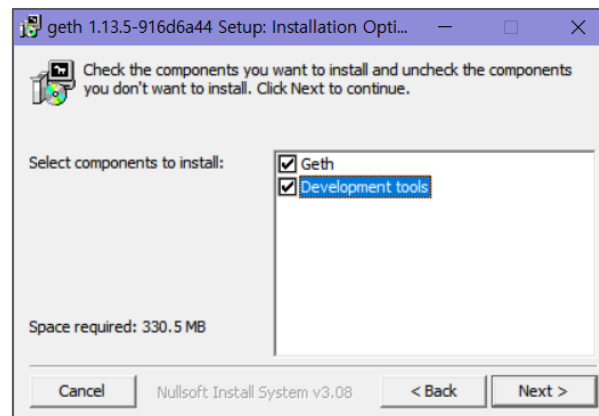
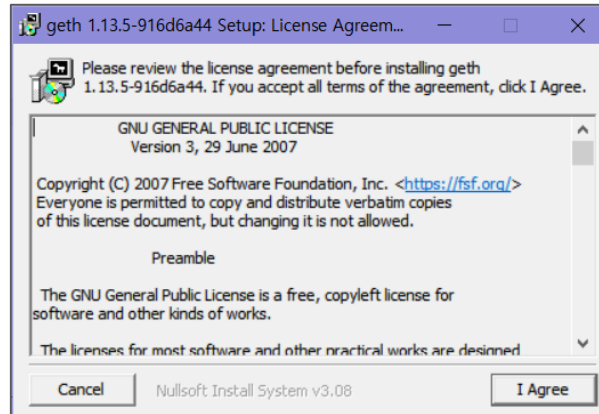
1. go-ethereum 테스트용 인스톨 툴 geth 아래 경로에서 최신 버전(FOR WINDOWS)을 다운로드 받습니다.

<https://geth.ethereum.org/downloads>

이 때 주의할 점은 개인 테스트용 사설 네트워크를 이용하여 서버 연동 및 마이닝 작업은 1.12.0 버전 까지만 지원하므로, 1.12.0 버전을 다운로드 받아 사용해야 합니다.

본 과정은 노드 구축까지 진행하므로 최신 버전으로 진행하시면 됩니다.

## 2. geth 를 설치하십시오.



3. 앞에서 만들었던 제네시스 블록을 복사하여 geth 경로에 붙여 넣습니다.
4. 이제 geth 를 실행하여 실습을 진행해 봅시다.

geth 가 설치된 경로에 cmd 창 또는 Powershell 창을 실행합니다.

블록체인을 저장할 경로를 mkdir 명령으로 만들어 줍니다.

```

관리자: Windows PowerShell
PS C:\Program Files\Geth> mkdir c:\my_blockchain

디렉터리: C:\

Mode                LastWriteTime         Length Name
----                -
d-----          2023-12-14 오후 3:38             my_blockchain

PS C:\Program Files\Geth> |

```

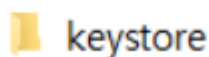
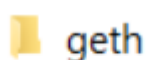
5. “geth --datadir <폴더명> init <genesis 블록 파일>” 을 실행합니다.

```

관리자: Windows PowerShell
PS C:\Program Files\Geth> ./geth --datadir c:\my_blockchain\ init c:\my_blockchain\genesis.json
INFO [12-14|15:44:46.570] Maximum peer count          ETH=50 LES=0 total=50
INFO [12-14|15:44:46.606] Set global gas cap          cap=50,000,000
INFO [12-14|15:44:46.606] Initializing the KZG library backend=gokzg
INFO [12-14|15:44:46.622] Defaulting to pebble as the backing database
INFO [12-14|15:44:46.622] Allocated cache and file handles database=c:\my_blockchain\geth\chaindata cache=16.00MiB handles=16
INFO [12-14|15:44:46.657] Opened ancient database      database=c:\my_blockchain\geth\chaindata\ancient\chain_readonly=false
INFO [12-14|15:44:46.657] State schema set to default  scheme=hash
INFO [12-14|15:44:46.657] Writing custom genesis block
INFO [12-14|15:44:46.659] Persisted trie from memory database nodes=1 size=150.00B time=1.5699ms gcnodes=0 gcsize=0.00B gctime=0s livenodes=0 livesize=0.00B
INFO [12-14|15:44:46.667] Successfully wrote genesis state database=chaindata hash=256305..6c5bda
INFO [12-14|15:44:46.667] Defaulting to pebble as the backing database database=c:\my_blockchain\geth\lightchaindata
INFO [12-14|15:44:46.667] Allocated cache and file handles ta cache=16.00MiB handles=16
INFO [12-14|15:44:46.691] Opened ancient database      database=c:\my_blockchain\geth\lightchaindata\ancient\chain_readonly=false
INFO [12-14|15:44:46.691] State schema set to default  scheme=hash
INFO [12-14|15:44:46.691] Writing custom genesis block
INFO [12-14|15:44:46.692] Persisted trie from memory database nodes=1 size=150.00B time=1.0092ms gcnodes=0 gcsize=0.00B gctime=0s livenodes=0 livesize=0.00B
INFO [12-14|15:44:46.700] Successfully wrote genesis state database=lightchaindata hash=256305..6c5bda
PS C:\Program Files\Geth> |

```

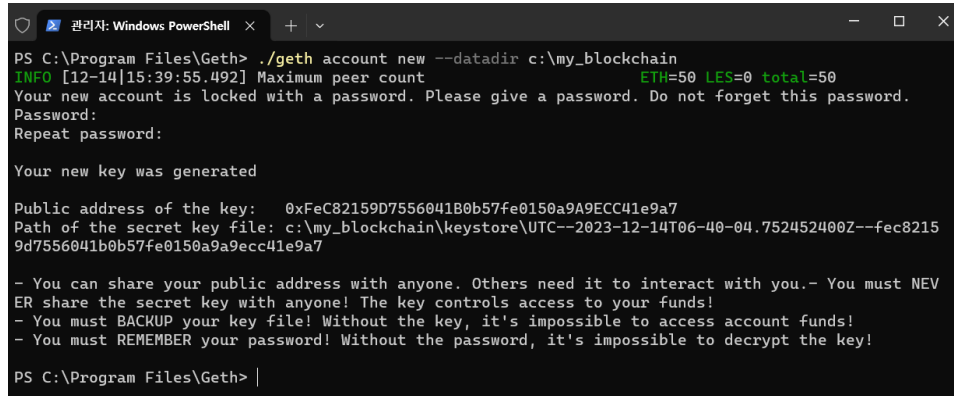
지정한 폴더에 아래와 같이 2 개의 폴더가 만들어져야 합니다.



- geth 폴더에는 모든 체인에 대한 데이터가 저장
- keystore 는 계정들에 대한 정보를 관리

6. `geth account new --datadir <폴더명>` 으로 계정을 생성해 줍니다.

이 때 비밀번호를 입력하라고 표시되며 2 번 비밀번호를 동일하게 입력하면 됩니다.



```

PS C:\Program Files\Geth> ./geth account new --datadir c:\my_blockchain
INFO [12-14|15:39:55.492] Maximum peer count ETH=50 LES=0 total=50
Your new account is locked with a password. Please give a password. Do not forget this password.
Password:
Repeat password:

Your new key was generated

Public address of the key: 0xFE82159D7556041B0b57fe0150a9A9ECC41e9a7
Path of the secret key file: c:\my_blockchain\keystore\UTC--2023-12-14T06-40-04.752452400Z--fec82159d7556041b0b57fe0150a9a9ecc41e9a7

- You can share your public address with anyone. Others need it to interact with you.- You must NEVER share the secret key with anyone! The key controls access to your funds!
- You must BACKUP your key file! Without the key, it's impossible to access account funds!
- You must REMEMBER your password! Without the password, it's impossible to decrypt the key!

PS C:\Program Files\Geth>
  
```

`geth account list --datadir <폴더명>`을 실행하면 생성된 계정 정보를 볼 수 있습니다.

## 6. 산출물 제출

---

1) 제출 위치:

<https://lab.ssafy.com/s10-study/self-project/> 의 “산출물 제출 가이드.docx” 참조