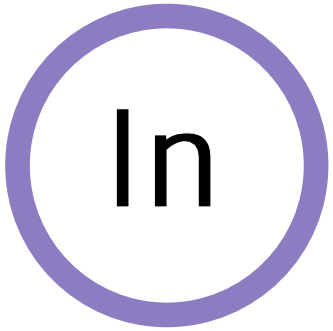
이더리움 DApp

Solidity 언어 실습



Solidity

07 DApp의

보안

대칭키, 비대칭키(암호화, 서명)

해시

PKI, SSL, TLS

네트워크

p2p

서버-클라이언트

DB

파일db

JSON

웹서비스

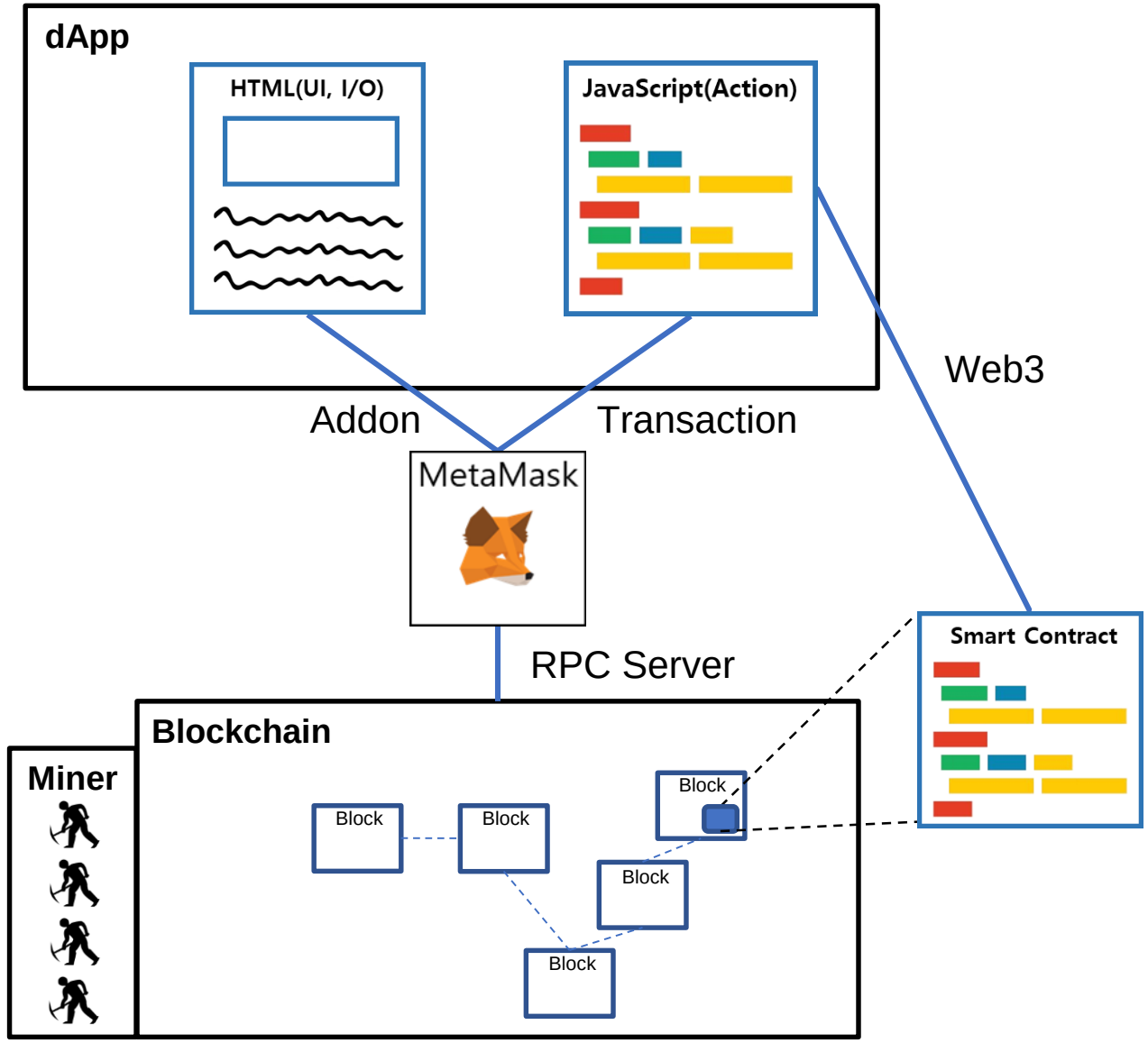
nodejs

반응형 웹페이지

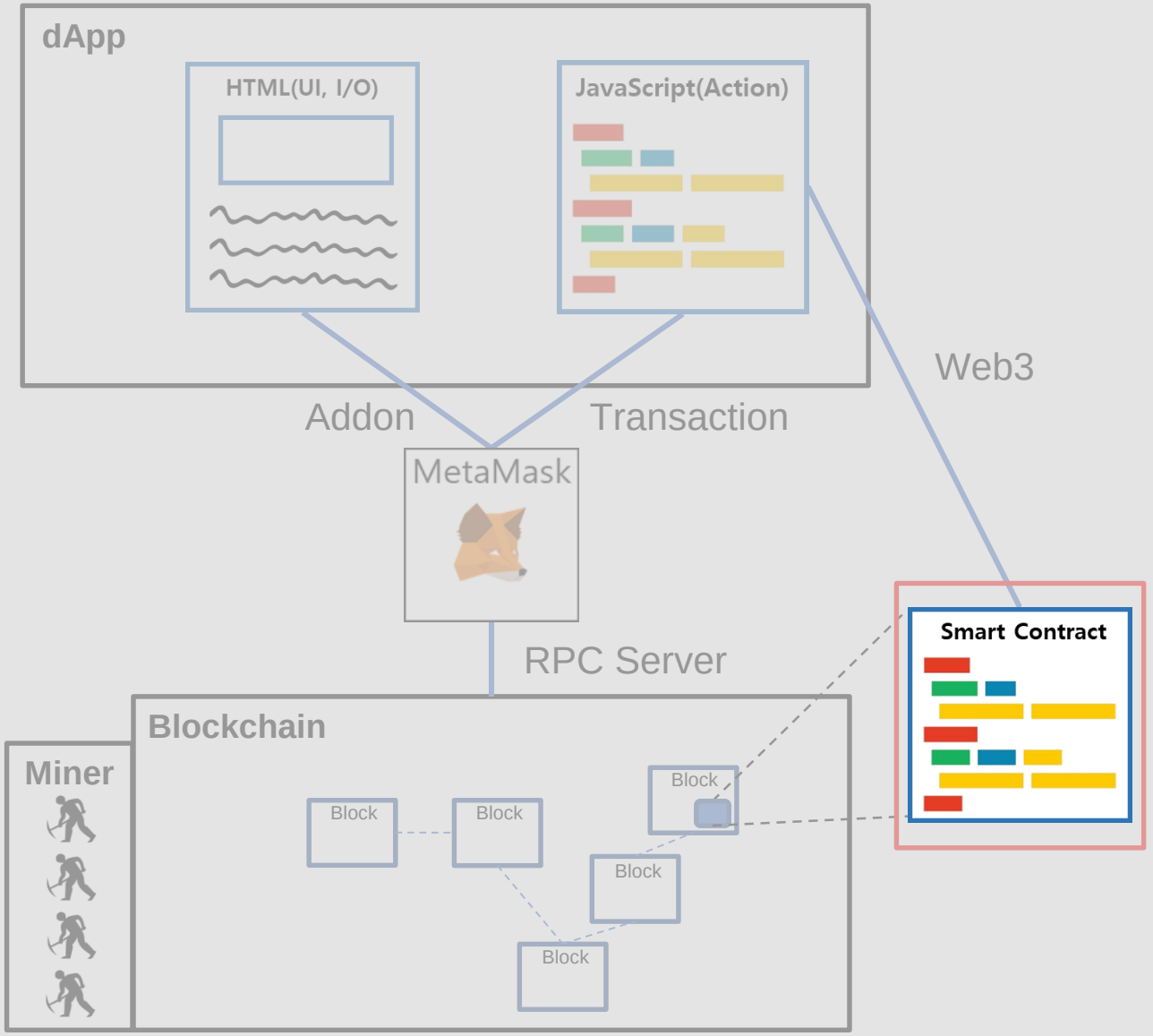
프로그래밍

스마트컨트랙트

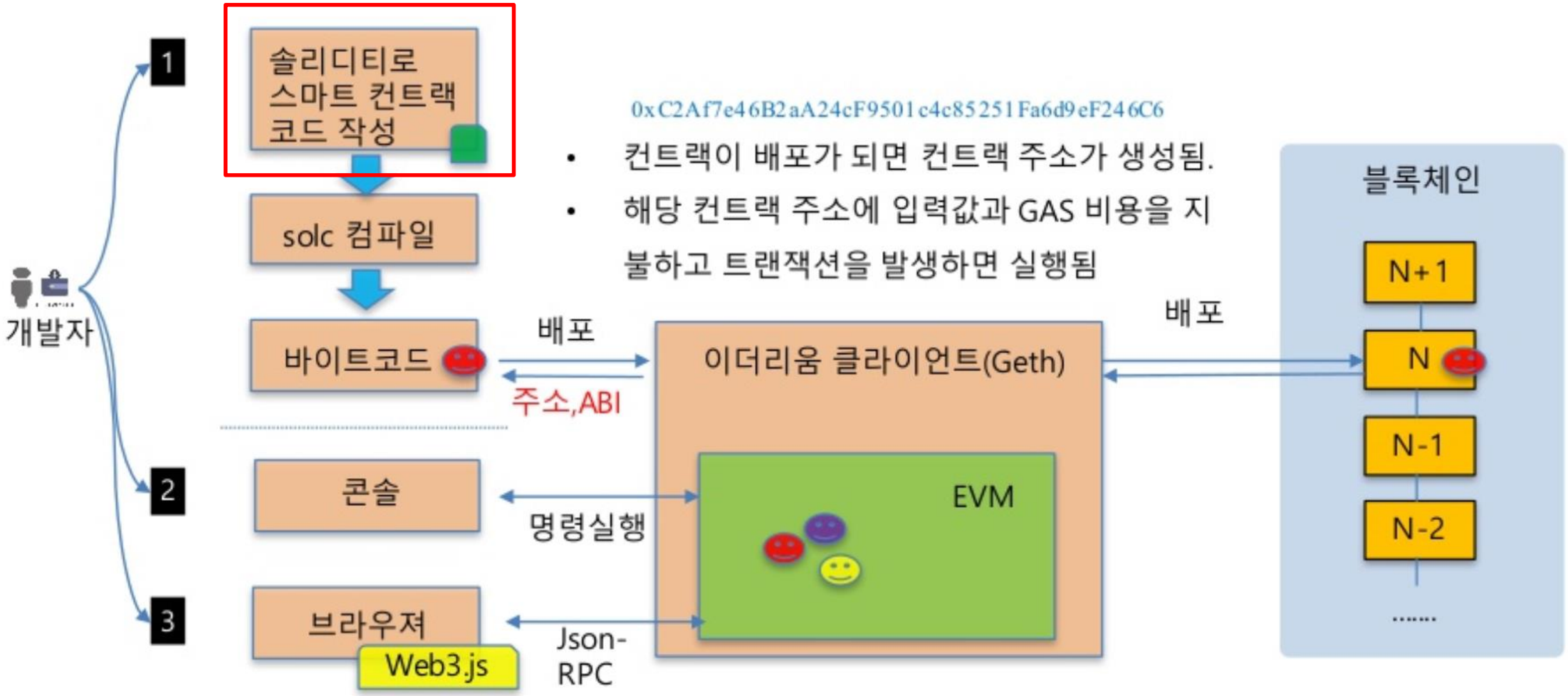
01 dApp Architecture



01 dApp Architecture

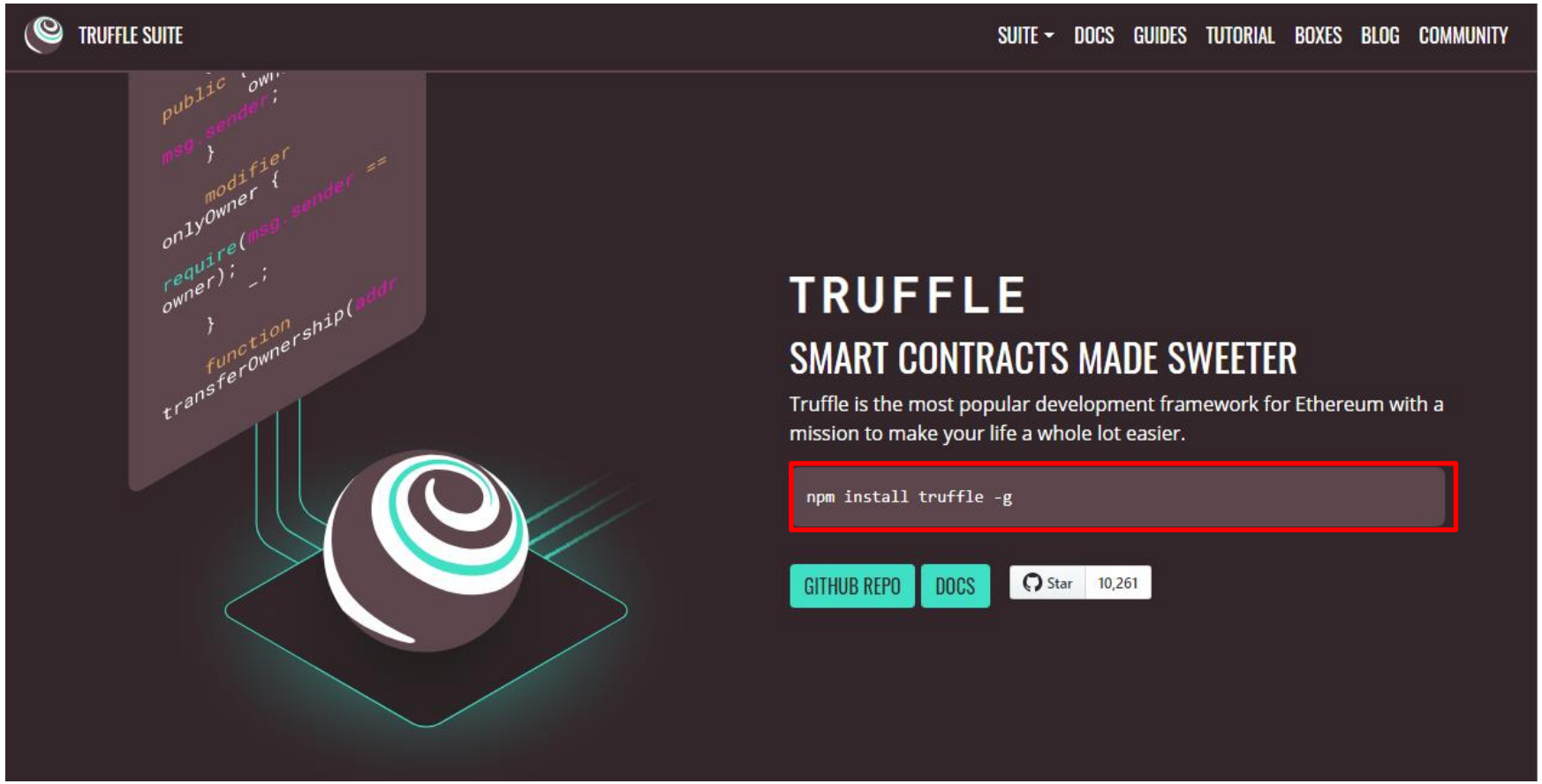


02 스마트 컨트랙트 구성요소



01 Truffle을 이용한 DApp

01 Truffle이란?



https://www.trufflesuite.com/

01 Truffle이란?

```
npm install truffle -g
```

npm?

npm: Node Packaged Manager

Node?

Node.js: **Javascript** runtime built on Chrome's V8 JavaScript engine

Javascript?

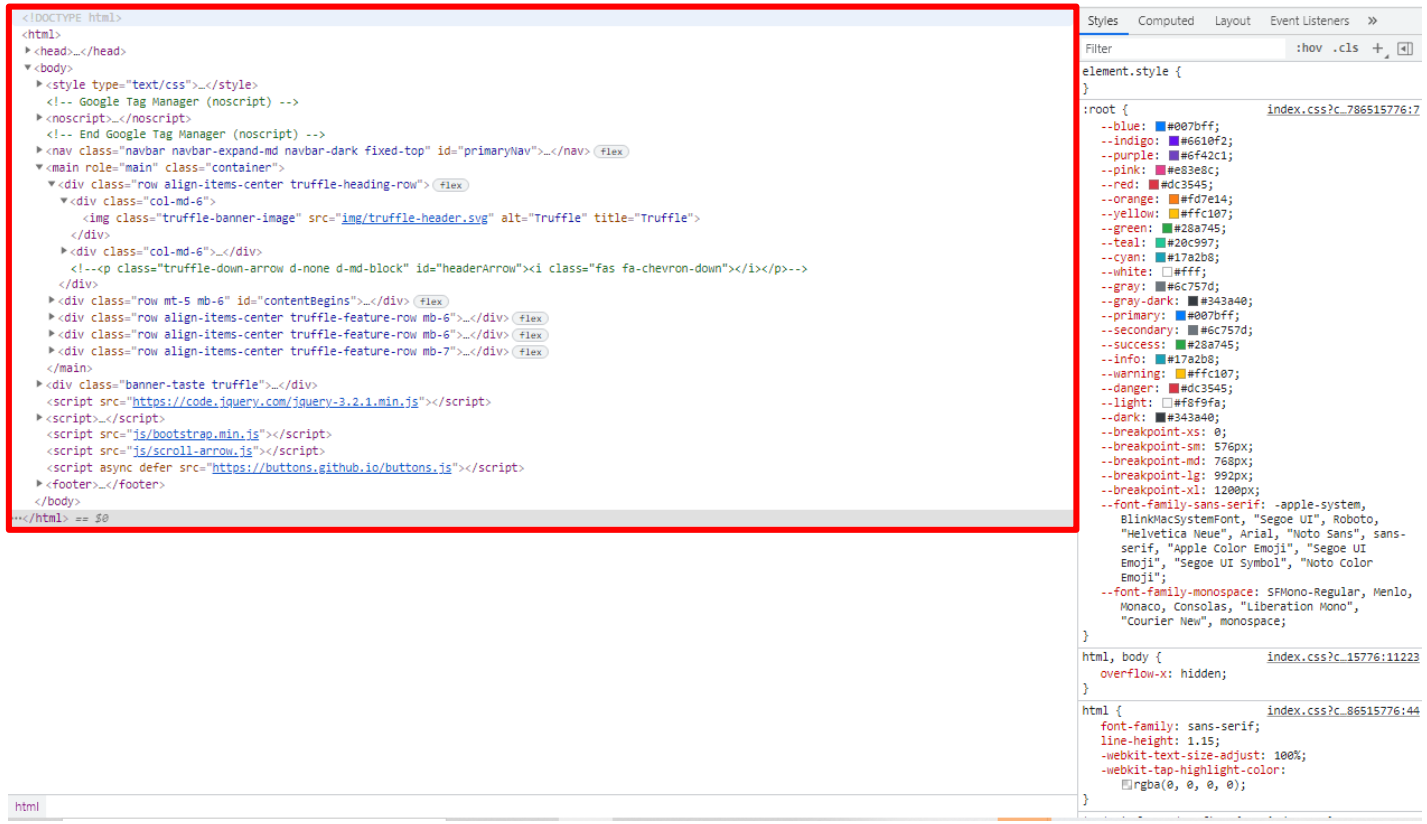
Javascript: 자바스크립트는 '웹페이지에 생동감을 불어넣기 위해' 만들어진 프로그래밍 언어. 동적인 DOM(HTML – 웹사이트의 표현을 위한 언어.)

01 Truffle이란?

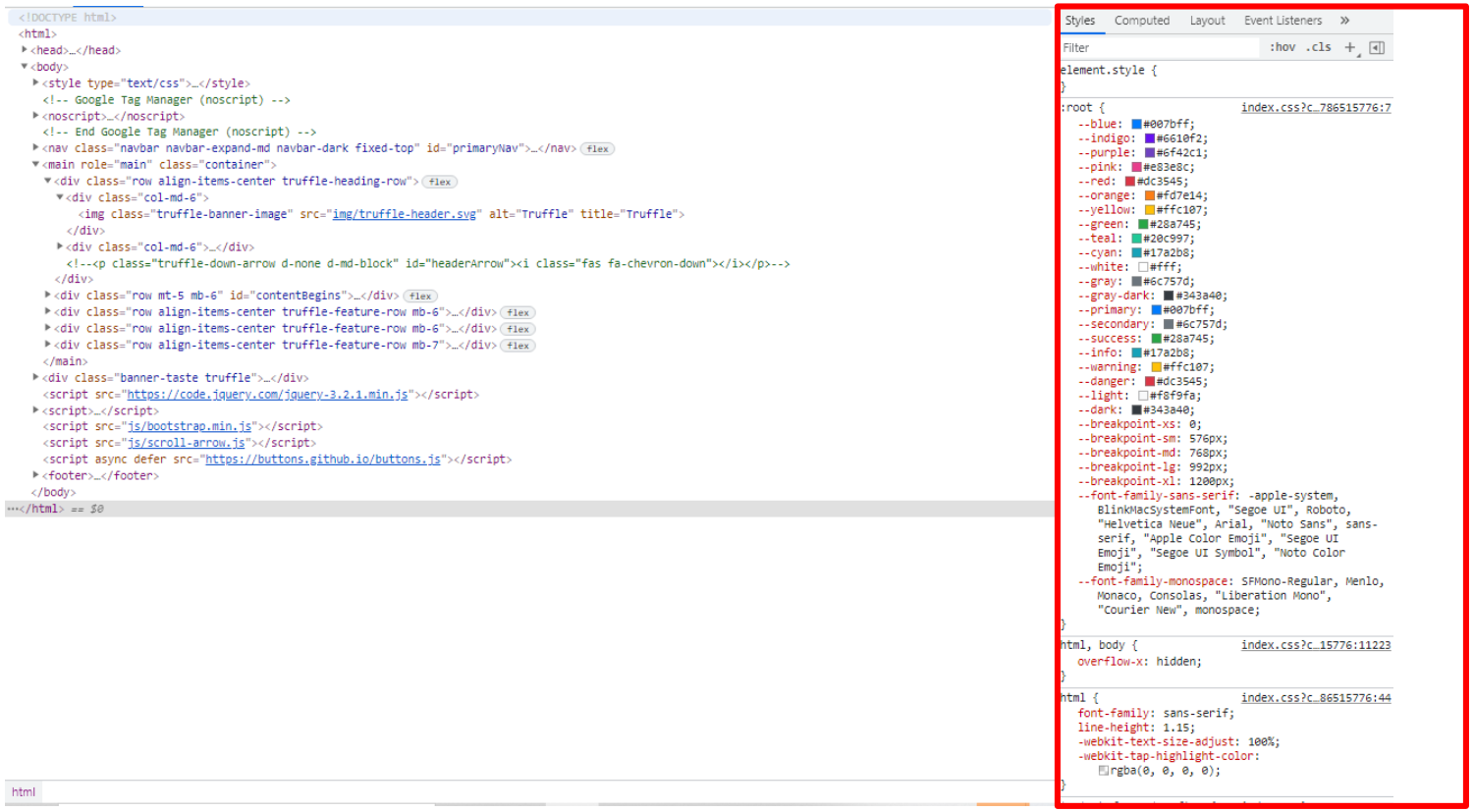
HTML?

HTML(Hyper Text Markup Language):

웹페이지를 만들기 위한 언어로 웹브라우저 위에서 동작하는 언어다.



01 Truffle이란?



CSS(Cascading Style Sheets: 상속적인 스타일 시트)
HTML의 스타일(어떻게 보이는지)를 지정해줌.

01 Truffle이란?

우리가 Truffle을 이용해 Dapp을 만든다는건?

- 블록체인과 통신하는 웹 application을 만든다는 것

그럼 알아야 하는 것은?

- Javascript + Node.js
- HTML
- CSS

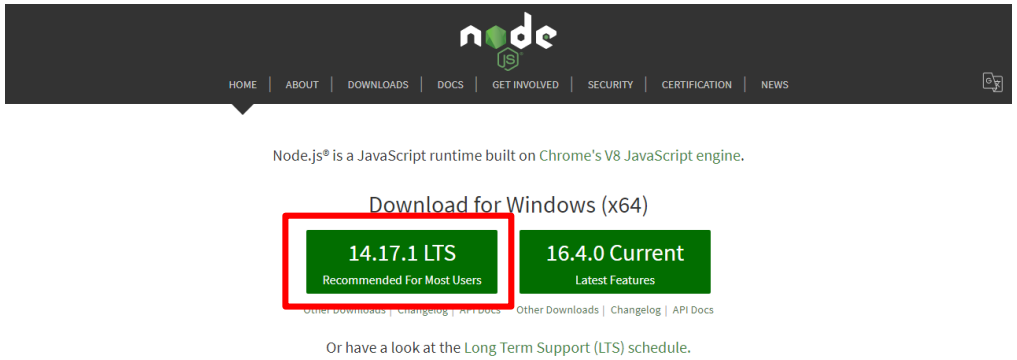
01 Truffle이란?

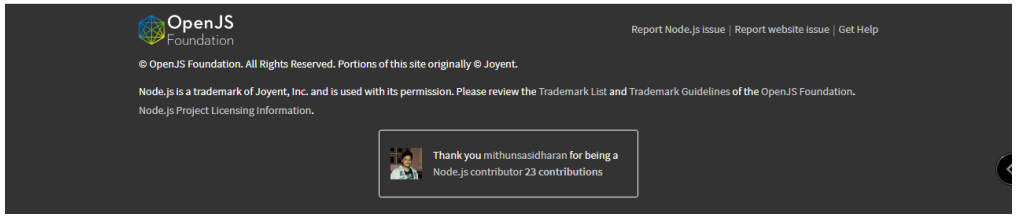
우선 개발환경을 설정해보자.

1. Node.js 설치
2. VS-Code Extension 설치
3. npm을 통한 truffle 전역설치
4. Metamask 설치 및 설정
5. ganache 설치

02 개발환경 설정

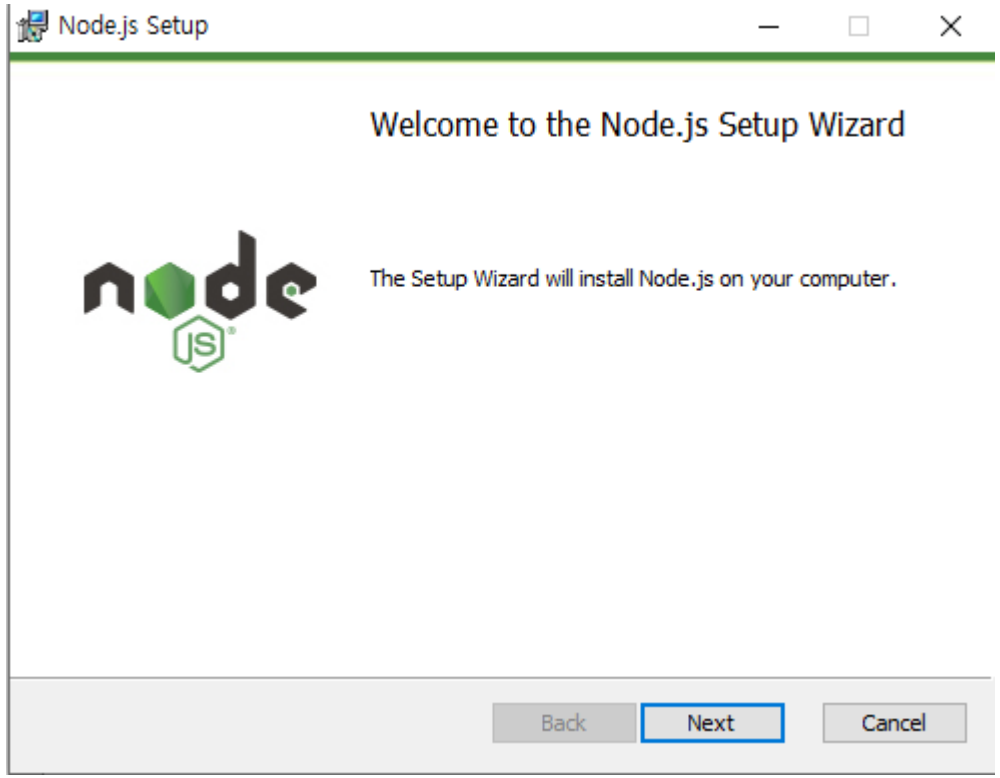
Node.js: **Javascript** runtime
built on Chrome’s V8 JavaScript engine
- https://nodejs.org/

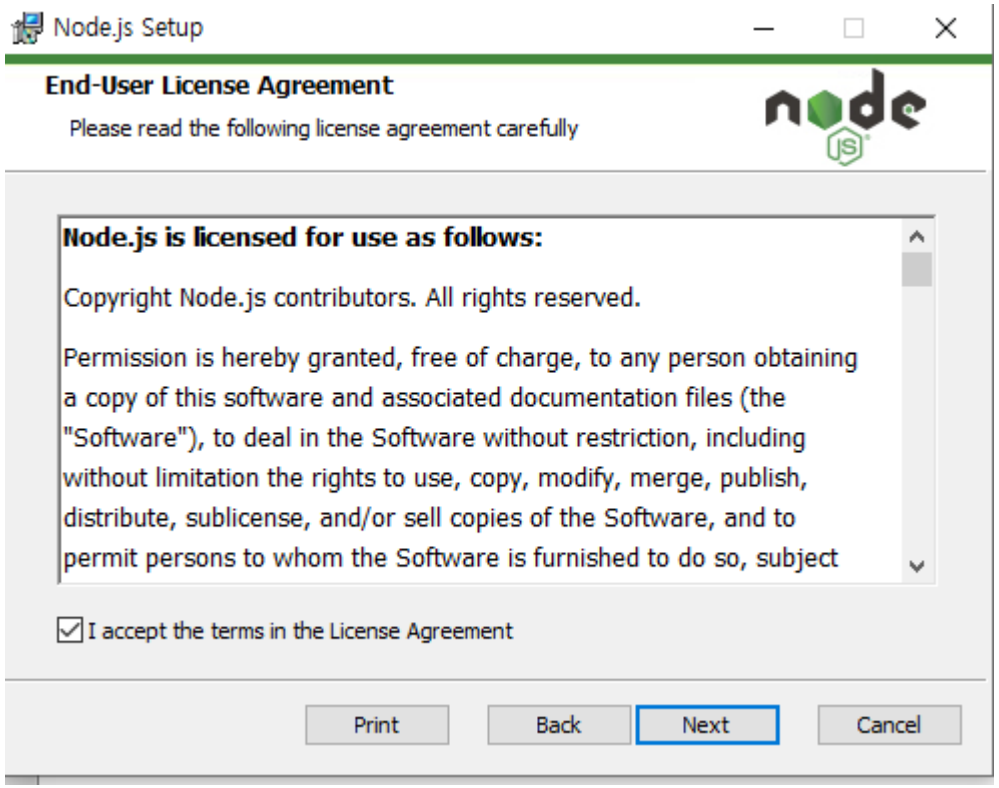




02 개발환경 설정

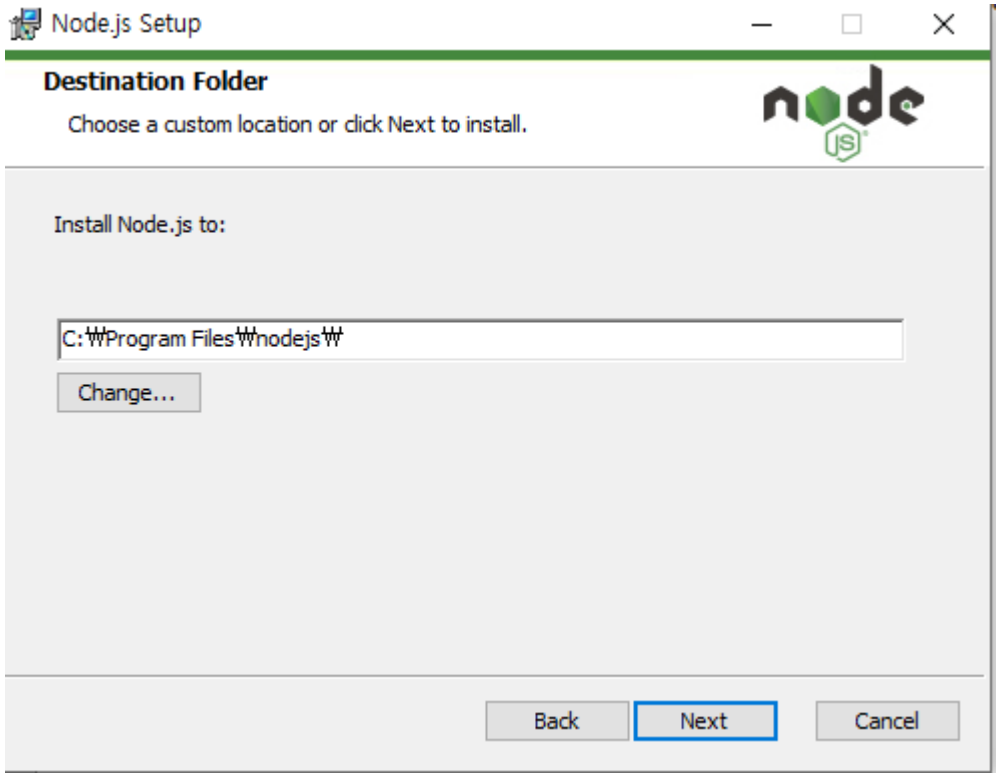
Node.js

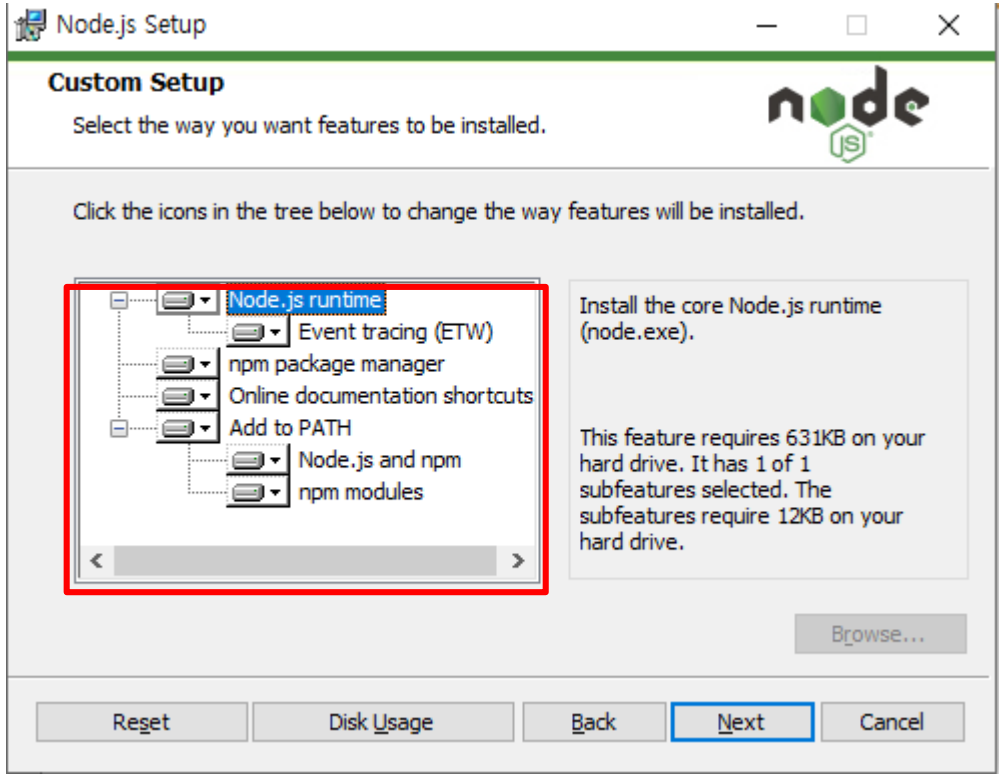




02 개발환경 설정

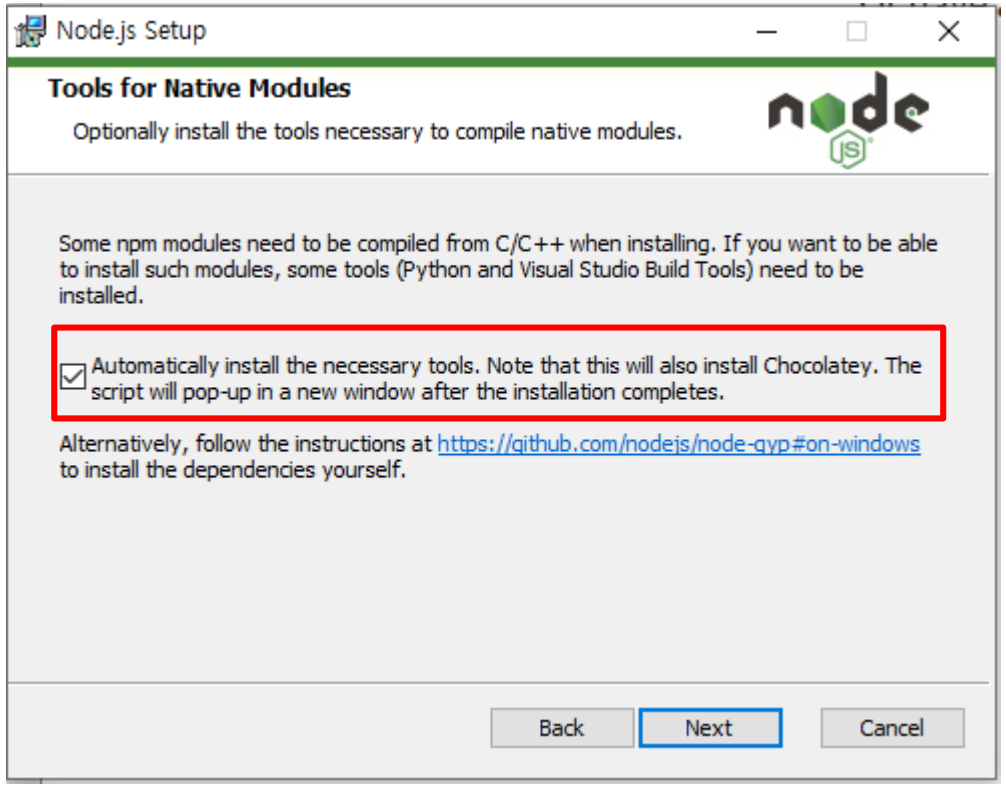
Node.js

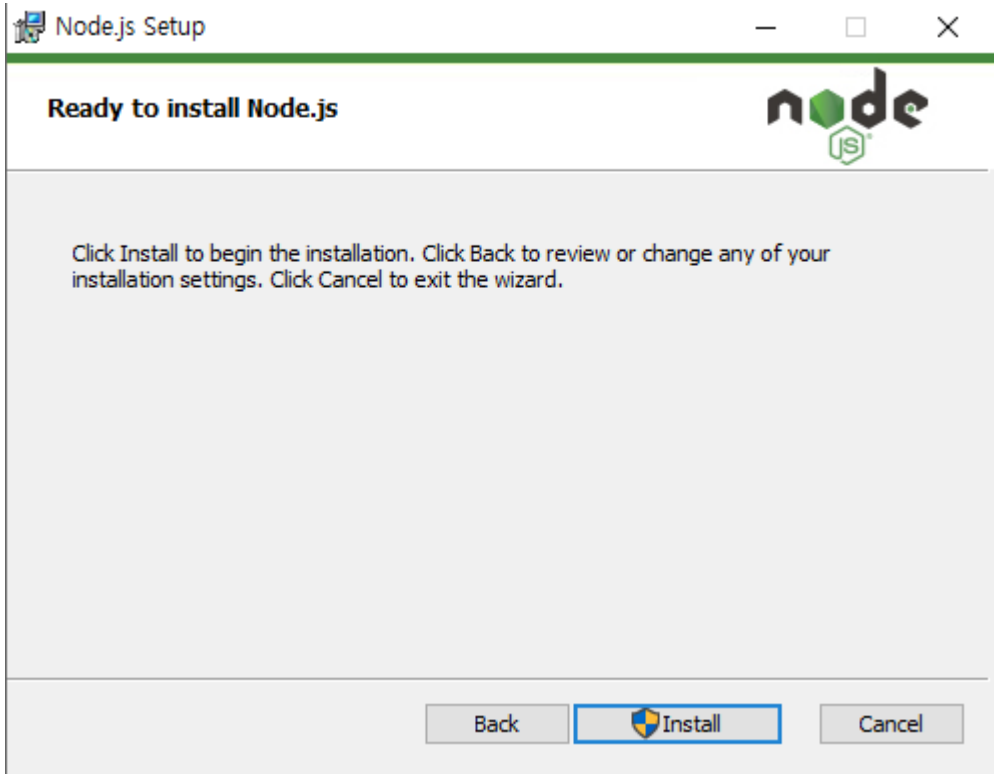




02 개발환경 설정

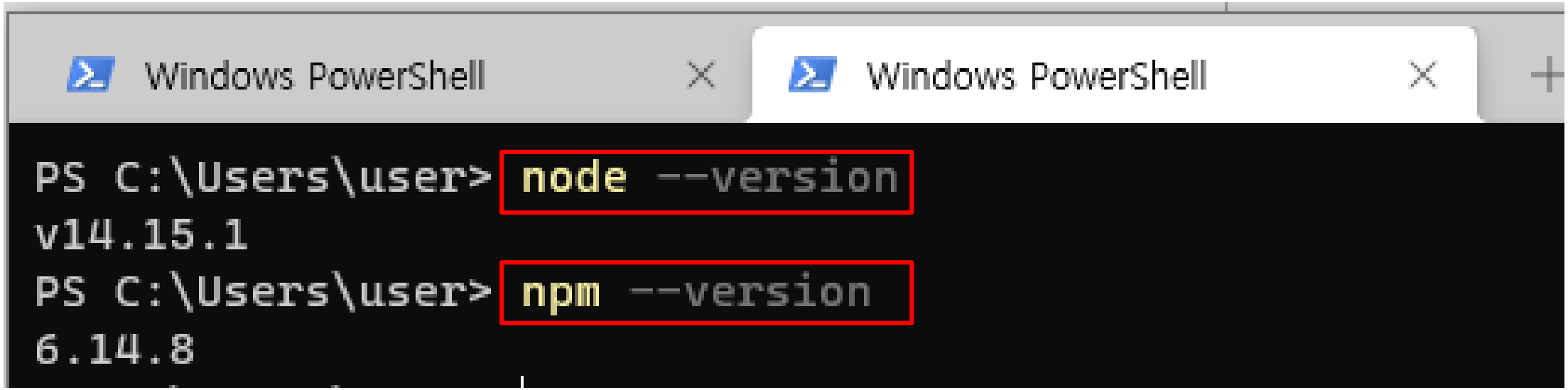
Node.js





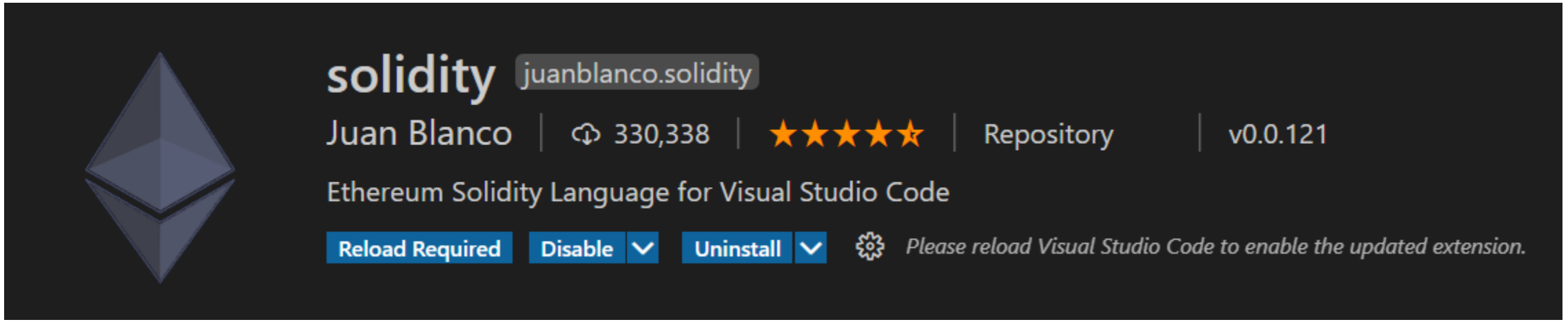
02 개발환경 설정

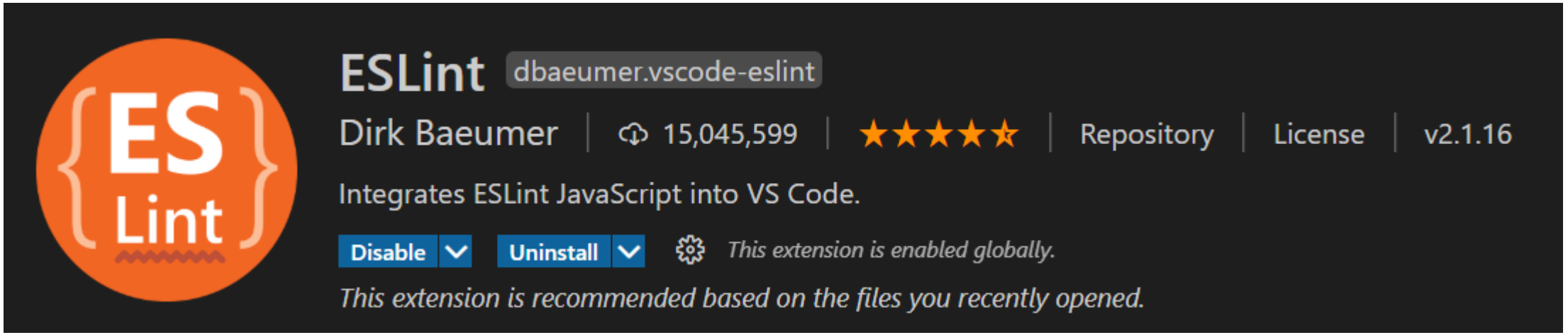
Node.js



02 개발환경 설정

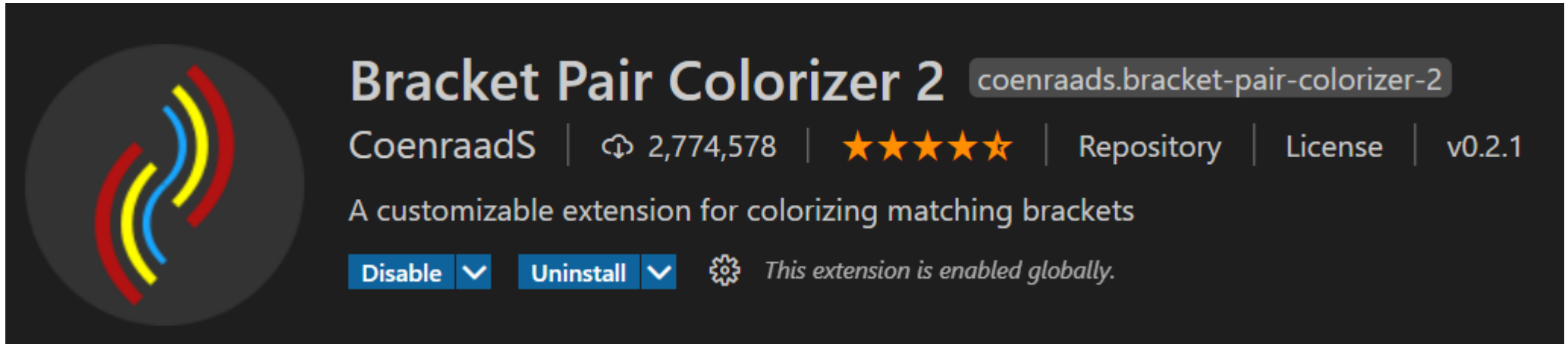
VS-Code Extension 설치

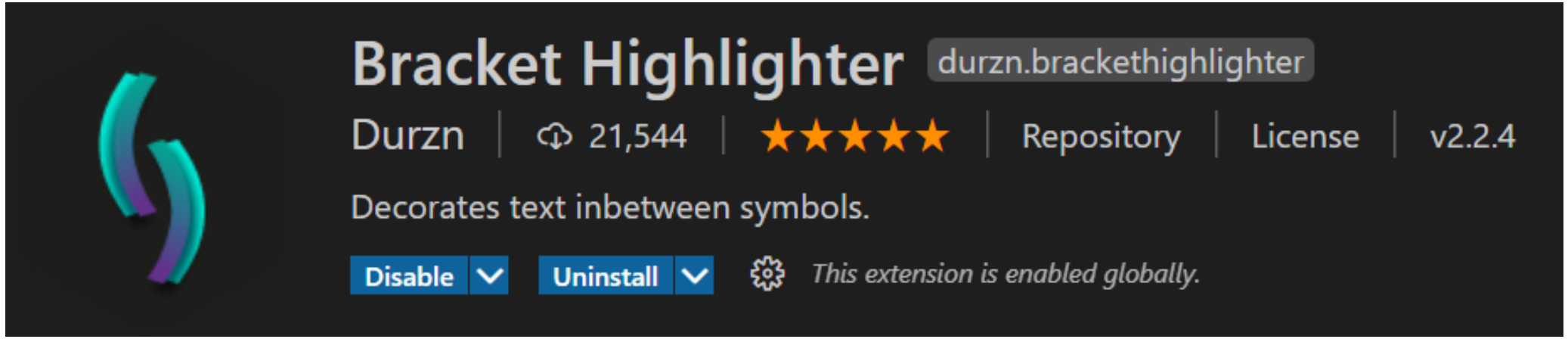




02 개발환경 설정

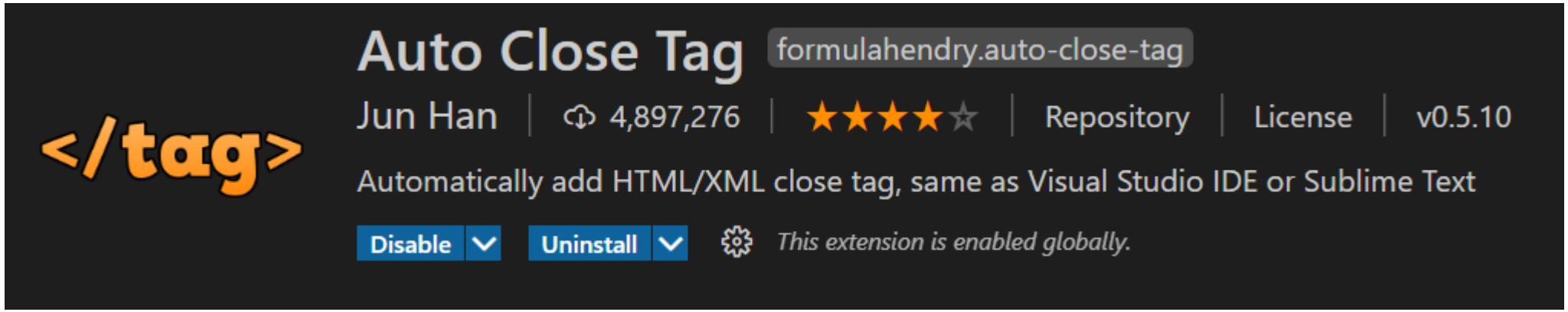
VS-Code Extension 설치

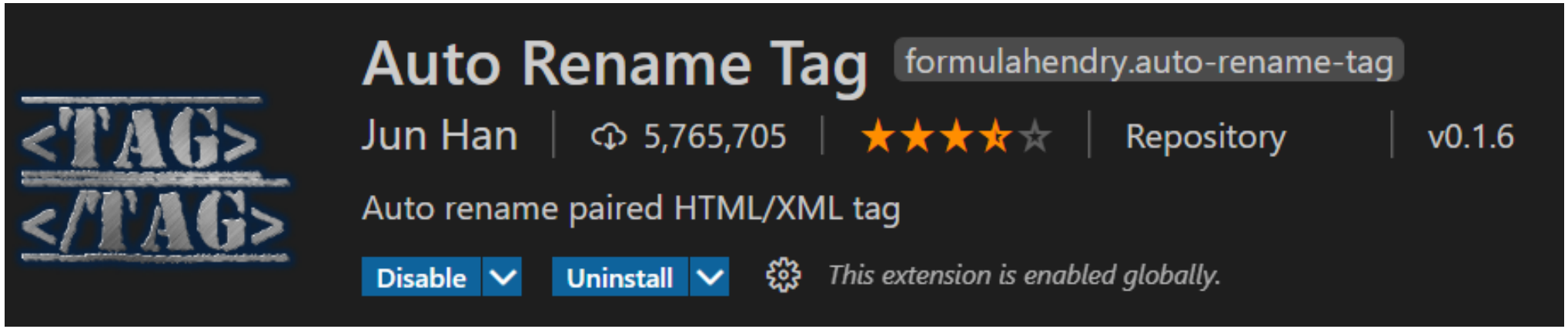




02 개발환경 설정

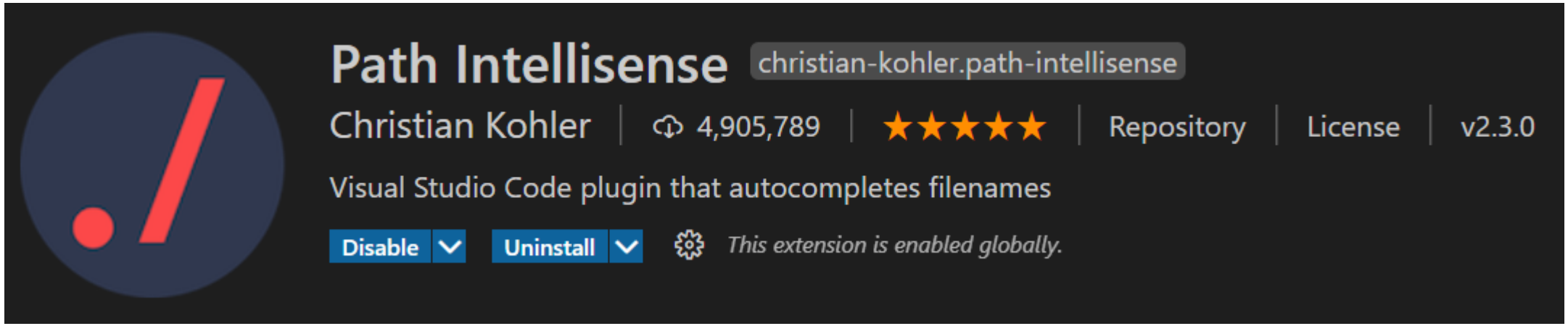
VS-Code Extension 설치

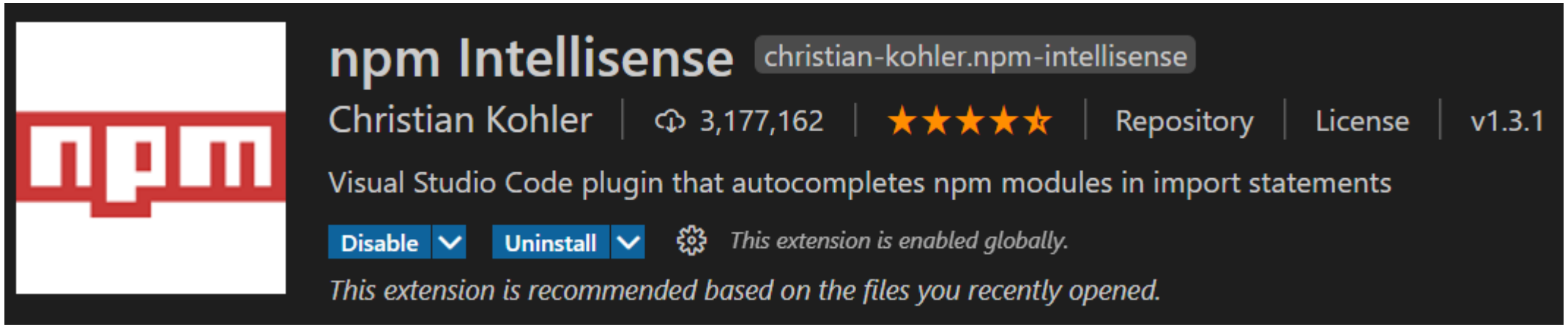




02 개발환경 설정

VS-Code Extension 설치





02 개발환경 설정

npm을 이용한 truffle 설치(전역 - global)

npm으로 패키지를 설치하려면?

→ npm install <패키지명>

npm으로 패키지를 전역으로 설치하려면?
→ npm install –g <패키지명>

```
PS C:\Users\user> npm install -g truffle
```

02 개발환경 설정

npm을 이용한 truffle 설치(전역 - global)

설치가 되면 truffle 명령어를 이용할 수 있다.

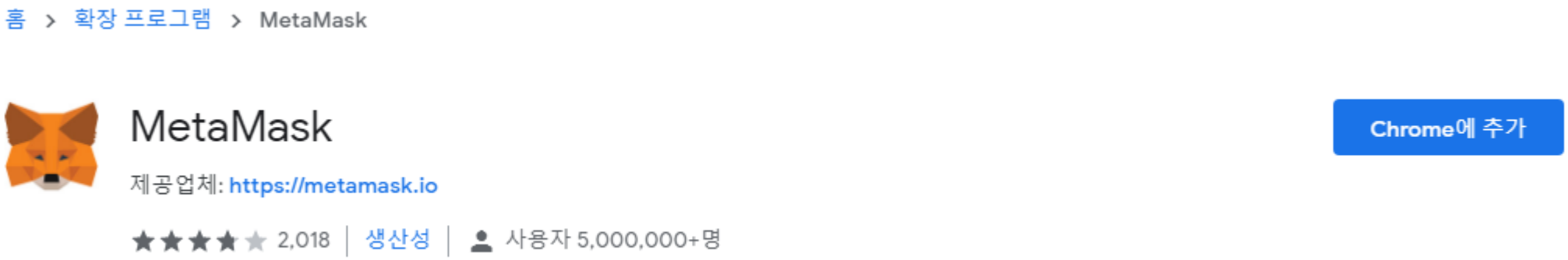
[주요 truffle 명령어]

- truffle init : truffle 프로젝트 생성
- truffle create : contract를 만들기 위한 헬퍼 명령어
- truffle compile : solidity 컴파일
- truffle migrate : solidity deploy.
- truffle develop : develop version **ethereum client** 실행.
- truffle install : solidity 외부 라이브러리 설치.

02 개발환경 설정

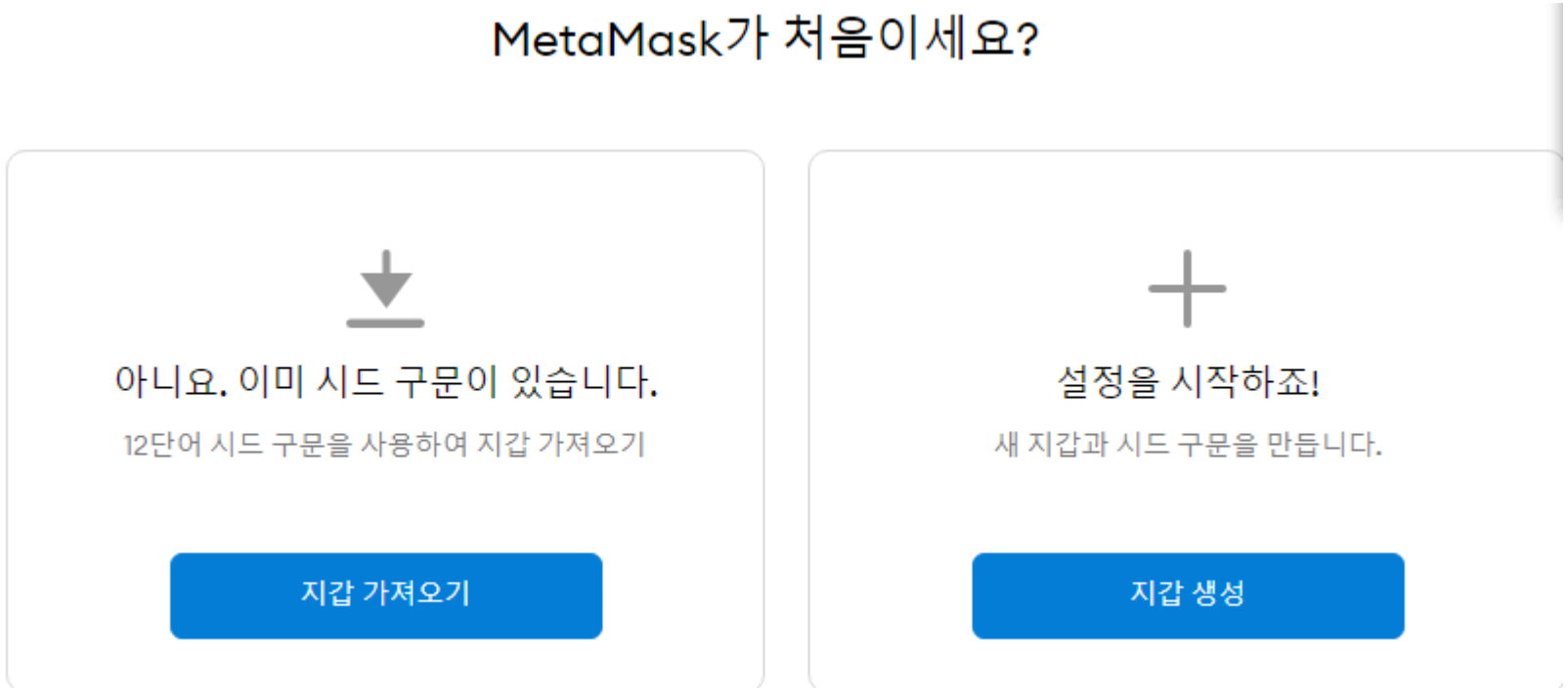
Metamask 설치

https://chrome.google.com/webstore/search/metamask?hl=ko



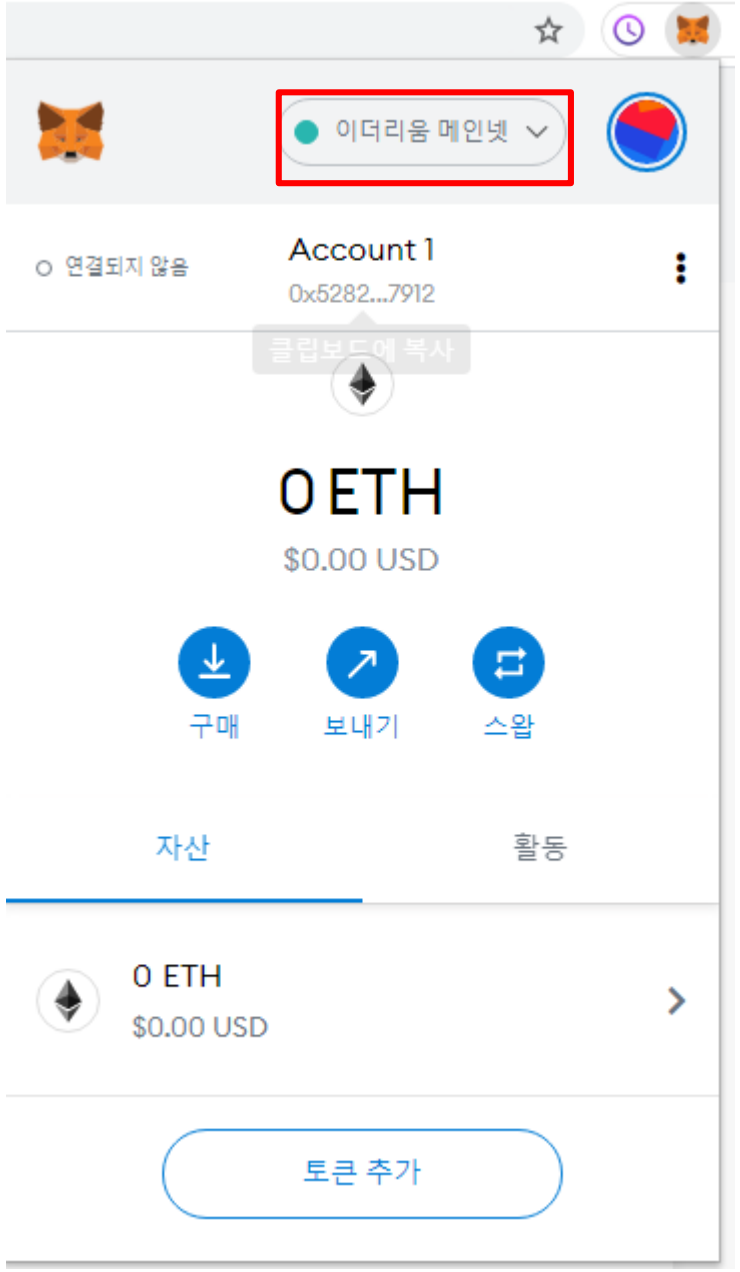
02 개발환경 설정

Metamask 설치



02 개발환경 설정

Metamask 설치



02 개발환경 설정

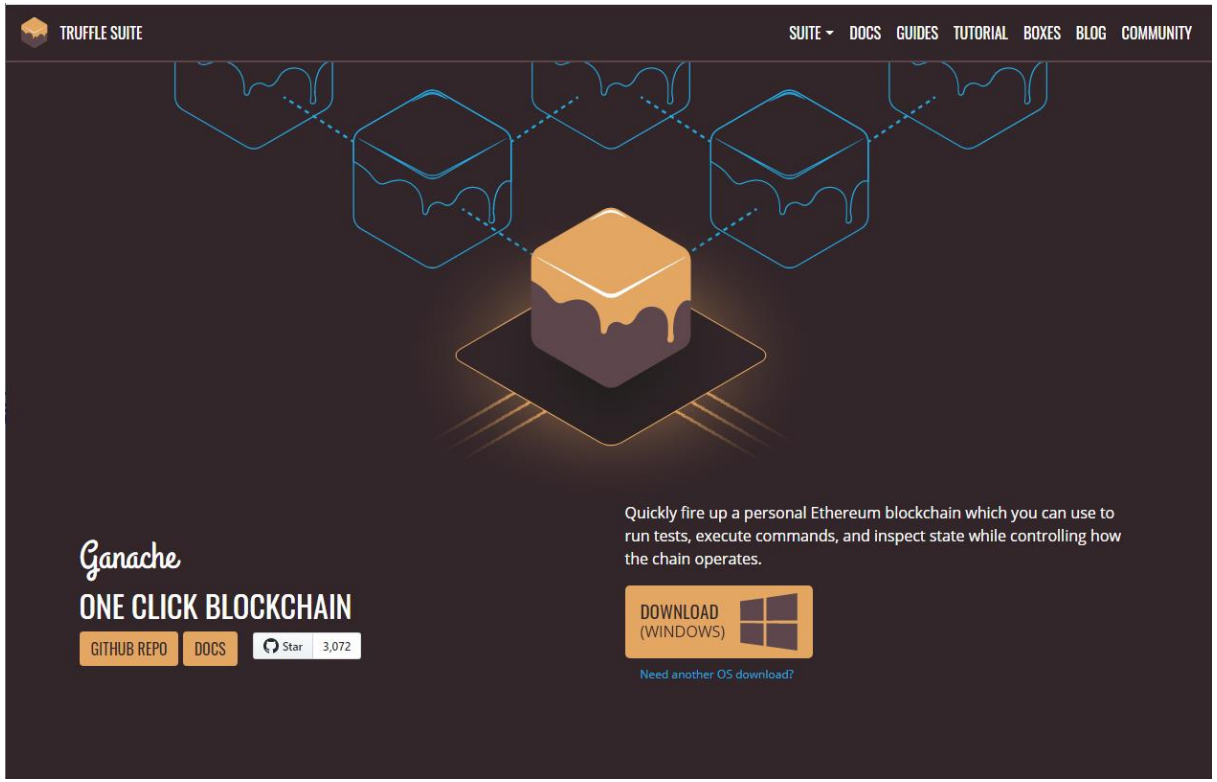
ganache 설치

https://www.trufflesuite.com/ganache

Quickly fire up a personal Ethereum blockchain which you can use to run tests, execute commands, and inspect state while controlling how the chain operates.

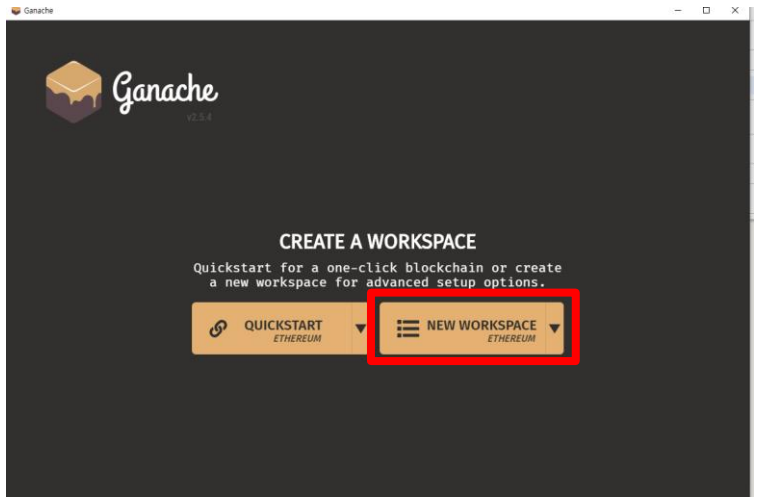
빠르게 개인 이더리움 블록체인을 시작하는 방법.

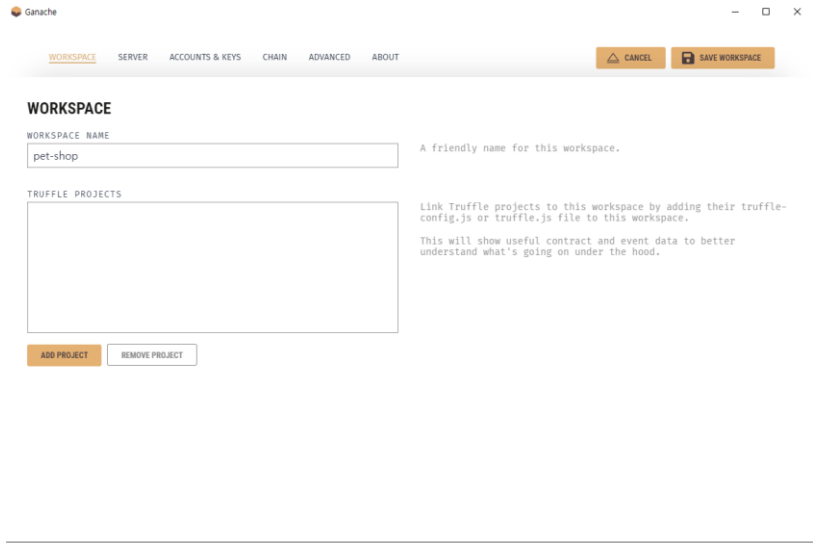
- contrract를 테스트 할 수 있는 gui 툴.



03 Solidity 개발 시작하기

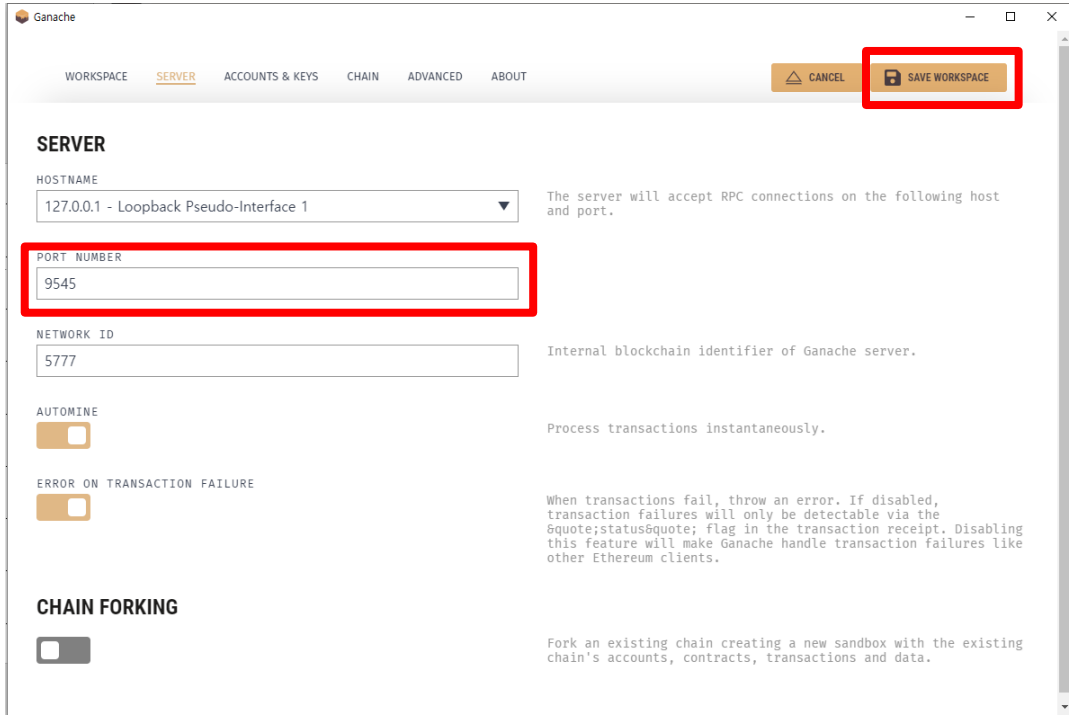
시작전에 ganache를 실행해 주세요.
- personal network를 실행하는 것





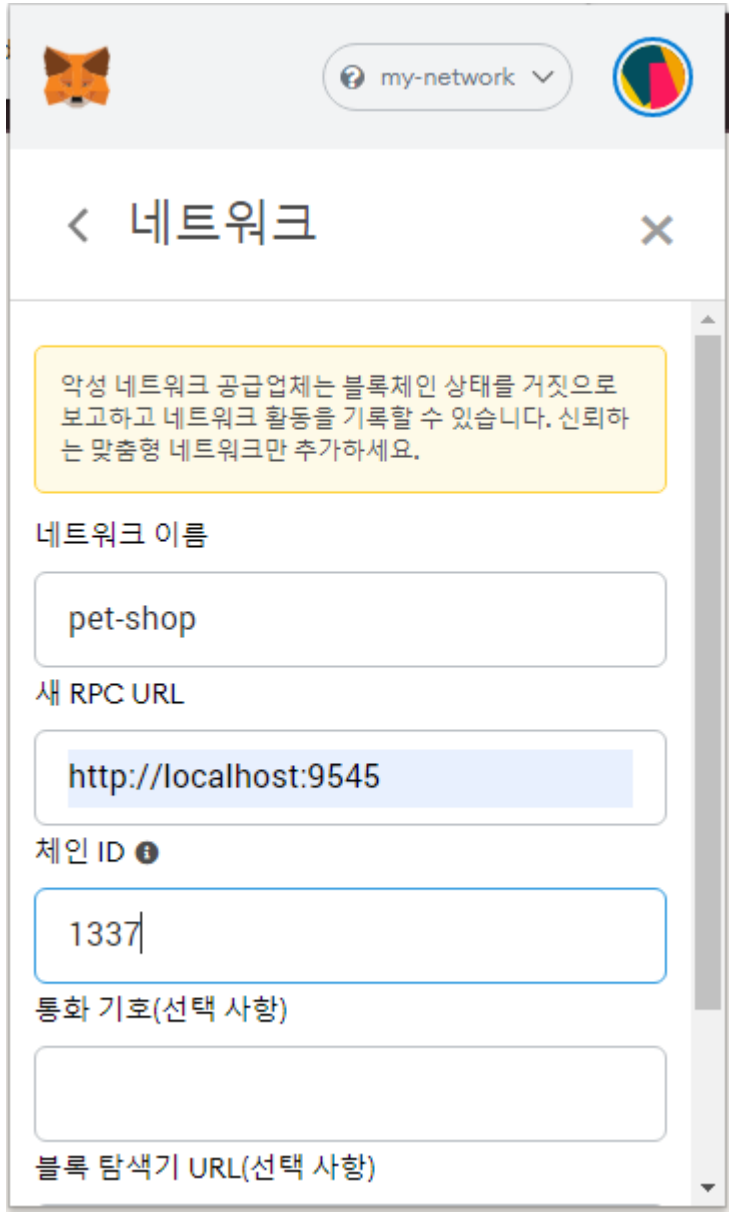
03 Solidity 개발 시작하기

시작전에 ganache를 실행해 주세요.
- personal network를 실행하는 것



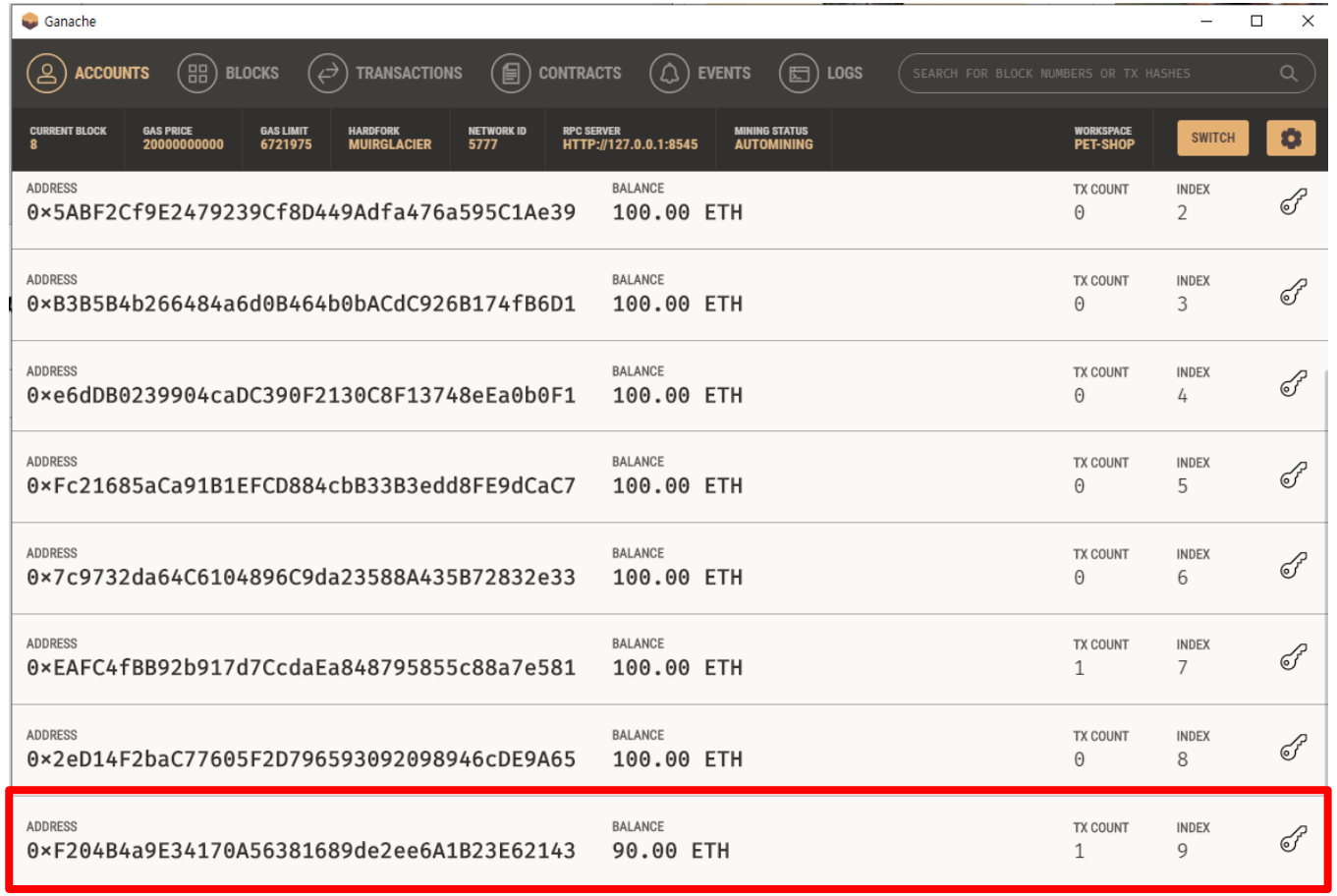
03 Solidity 개발 시작하기

metamask personal network로 설정



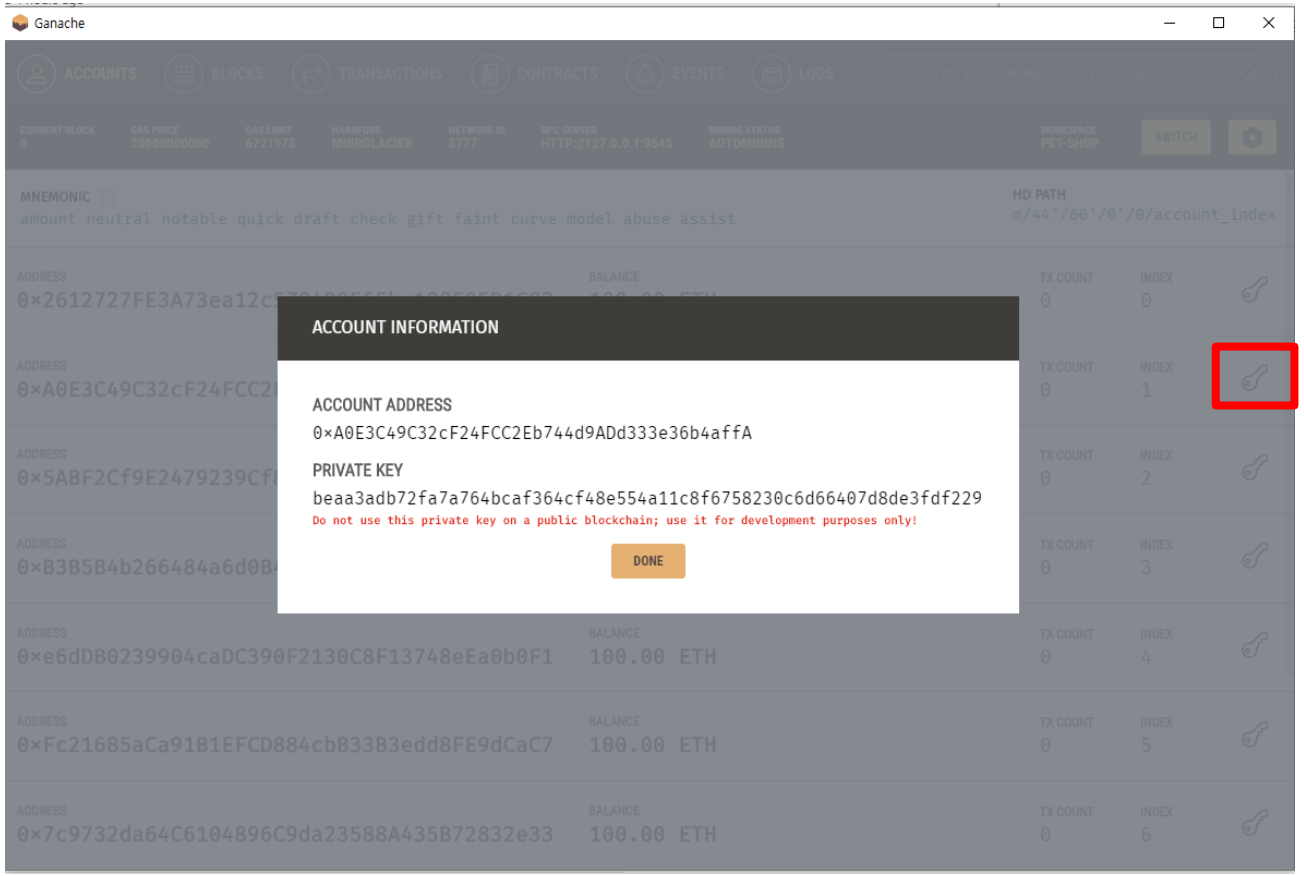
03 Solidity 개발 시작하기

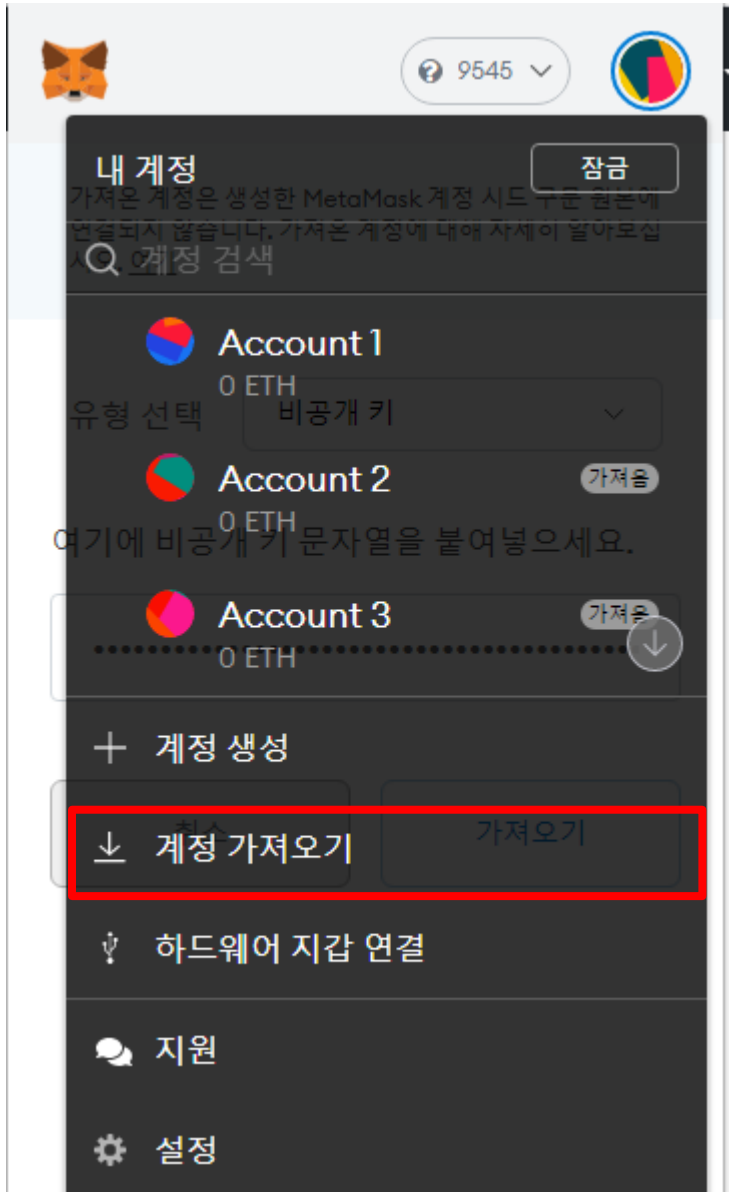
이더 보내기 실습



03 Solidity 개발 시작하기

Metamask 계정추가





03 Solidity 개발 시작하기

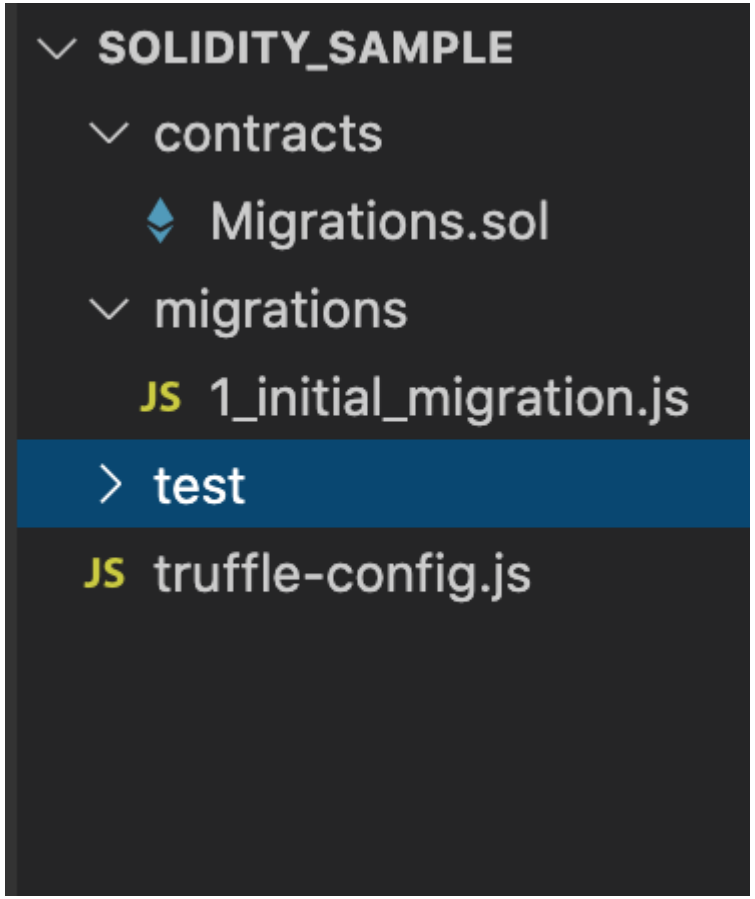
실습)

Metamask로 Personal Network
송금을 해보세요!

03 Solidity 개발 시작하기

truffle 환경에서 Contract 개발시작하기

```
> mkdir <project>
> cd <project>
> truffle init
```



03 Solidity 개발 시작하기

truffle 환경에서 Contract 개발시작하기

> truffle create contract <Contract 이름>

03 Solidity 개발 시작하기

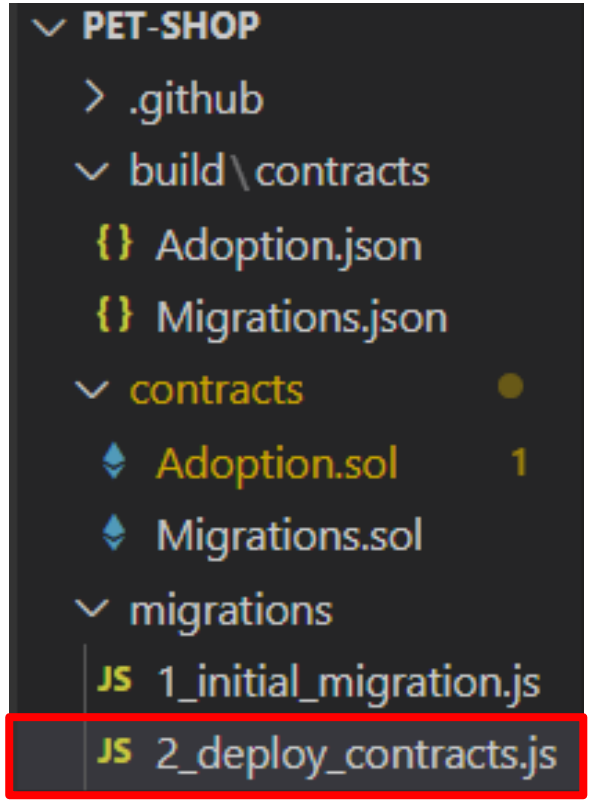
Contract 만들기

```
// SPDX-License-Identifier: MIT
pragma solidity >=0.4.22 <0.9.0;

contract Adoption {
  address[16] public adopters;

  constructor() public {
    // 생성자. (생성자 이름이 0.5버전 부터 변경)
  }

  function adopt(uint petId) public returns (uint){
    require(petId>=0 && petId <=15);

    adopters[petId] = msg.sender;
    return petId;
  }

  function getAdopters() external view returns (address[16] memory){
    return adopters;
  }
}
```

03 Solidity 개발 시작하기

Contract 만들기

migrations/2_deploy_contracts.js 생성




```
var Adoption = artifacts.require("Adoption");

module.exports = function(deployer) {
  deployer.deploy(Adoption);
};
```

03 Solidity 개발 시작하기

Contract 만들기

> truffle migrate

```
truffle(develop)> migrate

Compiling your contracts...
===========================
> Everything is up to date, there is nothing to compile.



Starting migrations...
======================
> Network name:    'develop'
> Network id:      5777
> Block gas limit: 6721975 (0x6691b7)


2_deploy_contracts.js
=====================

   Deploying 'Adoption'
   --------------------
   > transaction hash:    0x3446203c5018bb0c518294f1b205c13474d857b6c20b850cc11819593a71469d
   > Blocks: 0            Seconds: 0
   > contract address:    0xdCABcece7D4E40153ad5Fd29feC5d3bC6dC93FB8
   > block number:        3
   > block timestamp:     1609384909
   > account:             0x0113ED56b67fa01cE4eA08067700E0Bf7D1C964F
   > balance:             99.99123808
   > gas used:            203815 (0x31c27)
   > gas price:           20 gwei
   > value sent:          0 ETH
   > total cost:          0.0040763 ETH


   > Saving migration to chain.
   > Saving artifacts
   -------------------------------------
   > Total cost:          0.0040763 ETH


Summary
=======
> Total deployments:   1
> Final cost:          0.0040763 ETH
```

04 Solidity 개발 맛보기 (pet-shop)

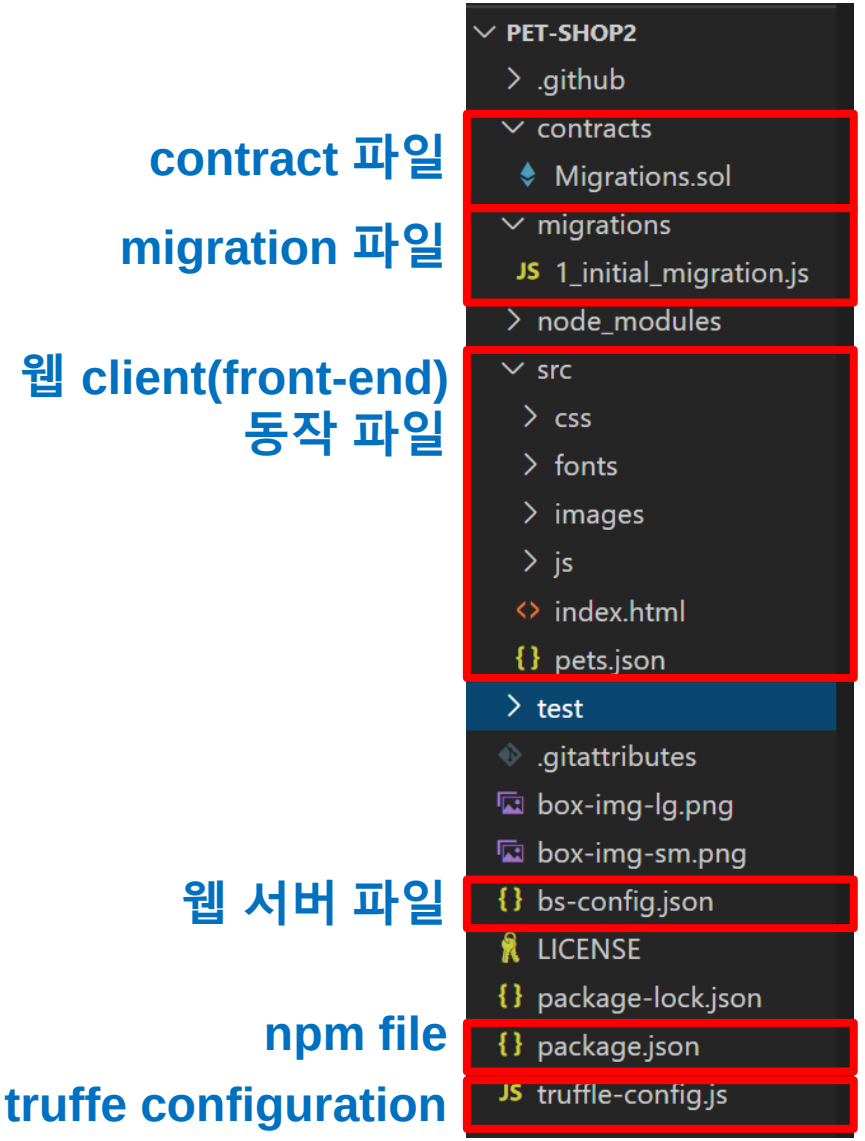
truffle pet-shop 실습하기

> truffle unbox pet-shop

```
Starting unbox...
=================

√ Preparing to download box
√ Downloading
npm WARN pet-shop@1.0.0 No description
npm WARN pet-shop@1.0.0 No repository field.
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.4 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.4: wanted {"os":"darwin","arch":"any
"} (current: {"os":"win32","arch":"x64"})

√ Cleaning up temporary files
√ Setting up box

Unbox successful, sweet!

Commands:

  Compile:        truffle compile
  Migrate:        truffle migrate
  Test contracts: truffle test
  Run dev server: npm run dev
```

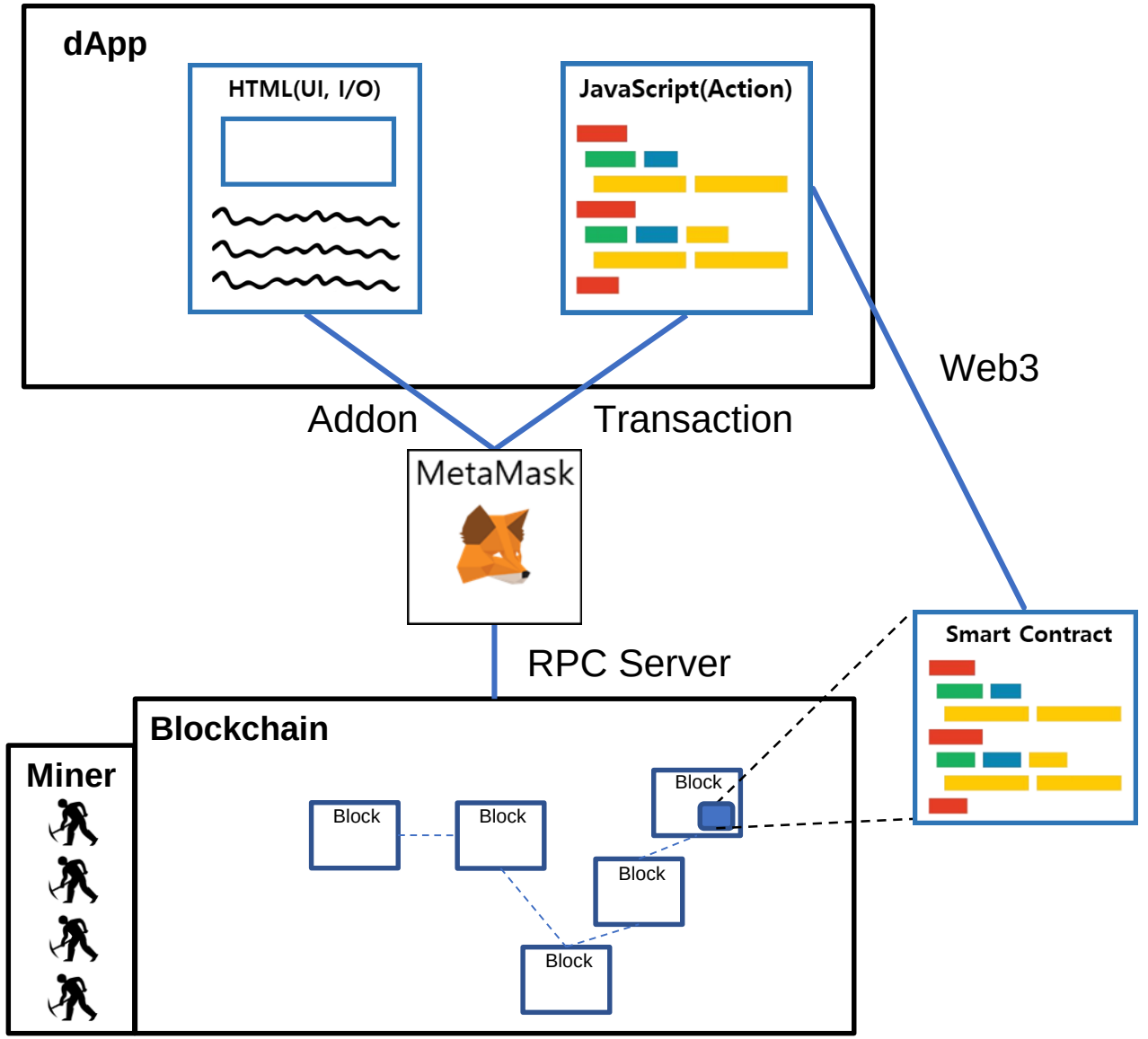
04 Solidity 개발 맛보기 (pet-shop)

truffle pet-shop 프로젝트 살펴보기



04 Solidity 개발 맛보기 (pet-shop)

Dapp Architecture

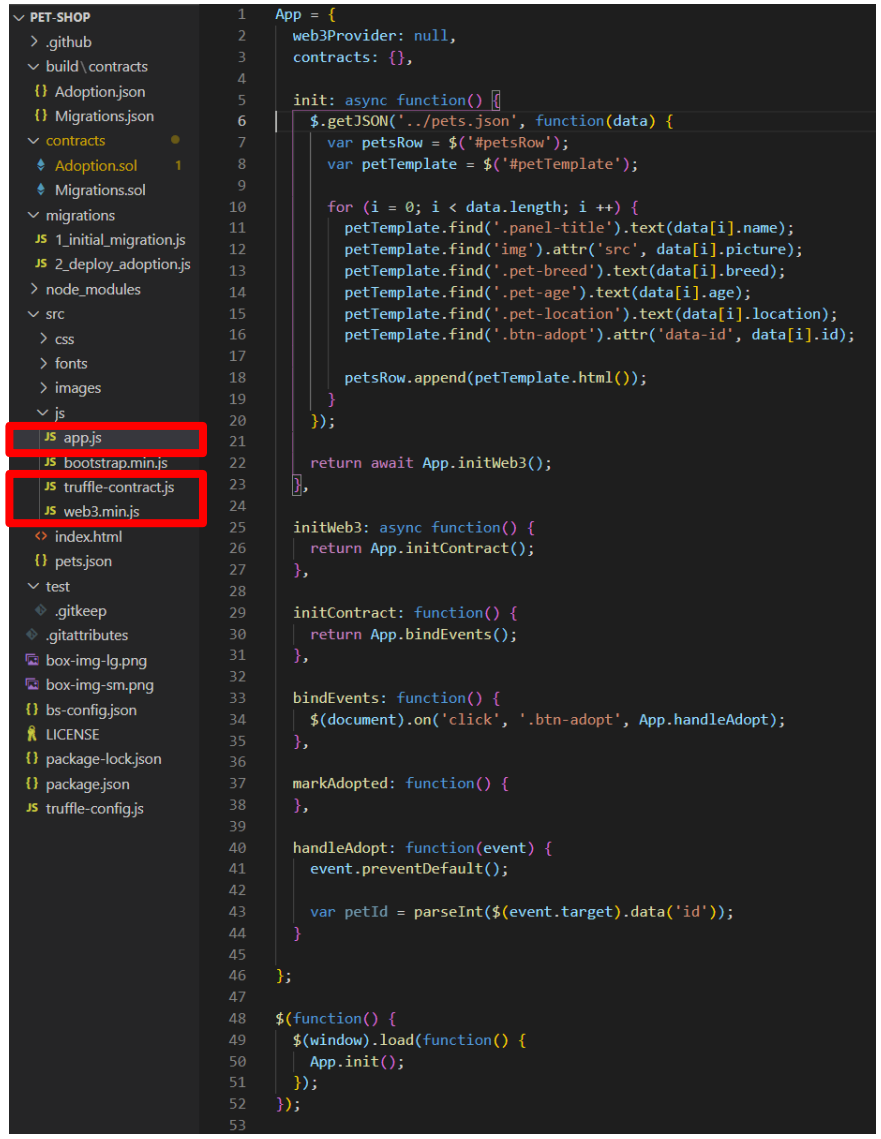


04 Solidity 개발 맛보기 (pet-shop)

app.js

우리는 우리의 Client(Front-End Application)이 블록체인 네트워크와 통신하기를 원한다.
➔ 그걸 도와주는 Front-End Library = Web3.js

truffle-contract.js도 web3.js 를 이용하고 있음.



04 Solidity 개발 맛보기 (pet-shop)

index.html

```
  <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
  <!-- Include all compiled plugins (below), or include individual files as needed -->
  <script src="js/bootstrap.min.js"></script>
  <script src="js/web3.min.js"></script>
  <script src="js/truffle-contract.js"></script>
  <script src="js/app.js"></script>
</body>
```

HTML 파일
➔ 해당 html파일이 서버를 실행시켰을 때 나오는 파일.

- jquery: js를 쉽게 사용하도록 도와주는 라이브러리
- bootstrap: css 라이브러리(스타일시트)

➔ 해당 파일에서 web3와 app.js 를 불러오는 것을 알 수 있음.

04 Solidity 개발 맛보기 (pet-shop)

서버 시작하기

> npm run dev

[package.json]

node.js라이브러리를 관리하기 위한 패키지 정의 파일

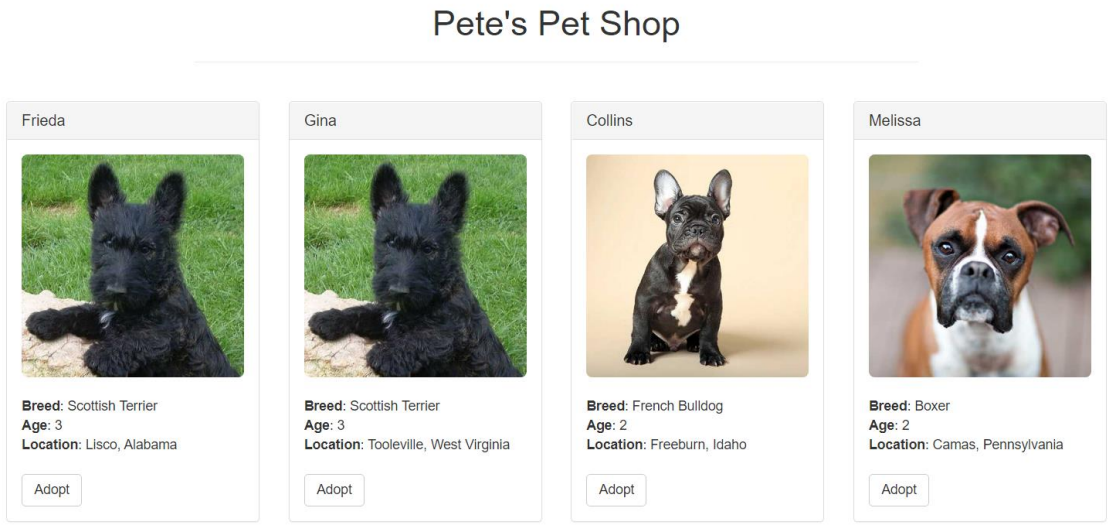
- scripts: “npm run <명령어>” 로 실행 가능함
- dependencies: 의존하는 라이브러리.
- devDependencies: 개발에서만 의존 라이브러리.

package.json

04 Solidity 개발 맛보기 (pet-shop)

서버 시작하기

> npm run dev



아직 어떤 것도 작동되지 않음.
➔ 아직 우린 js에서 통신하는 작업을 하지 않았기 때문.

04 Solidity 개발 맛보기 (pet-shop)

web3.js 적용하여 블록체인에 SmartContract 요청하기

[app.js] 수정

```
initWeb3: async function() {
  if (window.ethereum) {
    // metamask가 최근 설치된 Browser(크롬, firefox, edge ..)
    App.web3Provider = window.ethereum;
    try {
      // metamask 요청
      await window.ethereum.enable();
    } catch (error) {
      // account 요청 접근 제한
      alert("account 접근 제한")
    }
  }
  else if (window.web3) {
    // 구버전 Metamask
    App.web3Provider = window.web3.currentProvider;
  }
  else {
    // web3 인스턴스가 탐지가 안되었을 때..
    App.web3Provider = new Web3.providers.HttpProvider('http://localhost:8545');
  }
  web3 = new Web3(App.web3Provider);
  return App.initContract();
},
```

04 Solidity 개발 맛보기 (pet-shop)

web3.js 적용하여 블록체인에 SmartContract 요청하기

[app.js] 수정

```
49    initContract: function() {
50      $.getJSON('Adoption.json', function(data) {
51        // jquery 문법.
52        // Adoption.json을 가져온다 (abi가 포함된 artifact 파일)
53
54        const AdoptionArtifact = data; // 객체화
55
56        // truffle-contract.js 이용.
57        App.contracts.Adoption = TruffleContract(AdoptionArtifact);
58
59        // Web3 Provider 셋팅
60        App.contracts.Adoption.setProvider(App.web3Provider);
61
62        // 컨트랙트를 call 한다음. adopted된걸 마킹하기 위해 호출.
63        return App.markAdopted();
64      });
65
66      return App.bindEvents();
67    },
```

04 Solidity 개발 맛보기 (pet-shop)

web3.js 적용하여 블록체인에 SmartContract 요청하기

[app.js] 수정

```
73    markAdopted: function() {
74      let adoptionInstance;
75
76      // deploy 되어 있는지 확인한 후.
77      App.contracts.Adoption.deployed().then(function(instance) {
78        adoptionInstance = instance;
79
80        // Adoption Contract에서 getAdopters 를 호출. (조회하기.)
81        return adoptionInstance.getAdopters.call();
82      }).then(function(adopters) {
83        for (i = 0; i < adopters.length; i++) {
84          if (adopters[i] !== '0x0000000000000000000000000000000000000000') {
85            // 입양 된것은 입양 되었다고 표기하기
86            $('.panel-pet').eq(i).find('button').text('Success').attr('disabled', true);
87          }
88        }
89      }).catch(function(err) {
90        alert(err.message);
91      });
92    },
```

04 Solidity 개발 맛보기 (pet-shop)

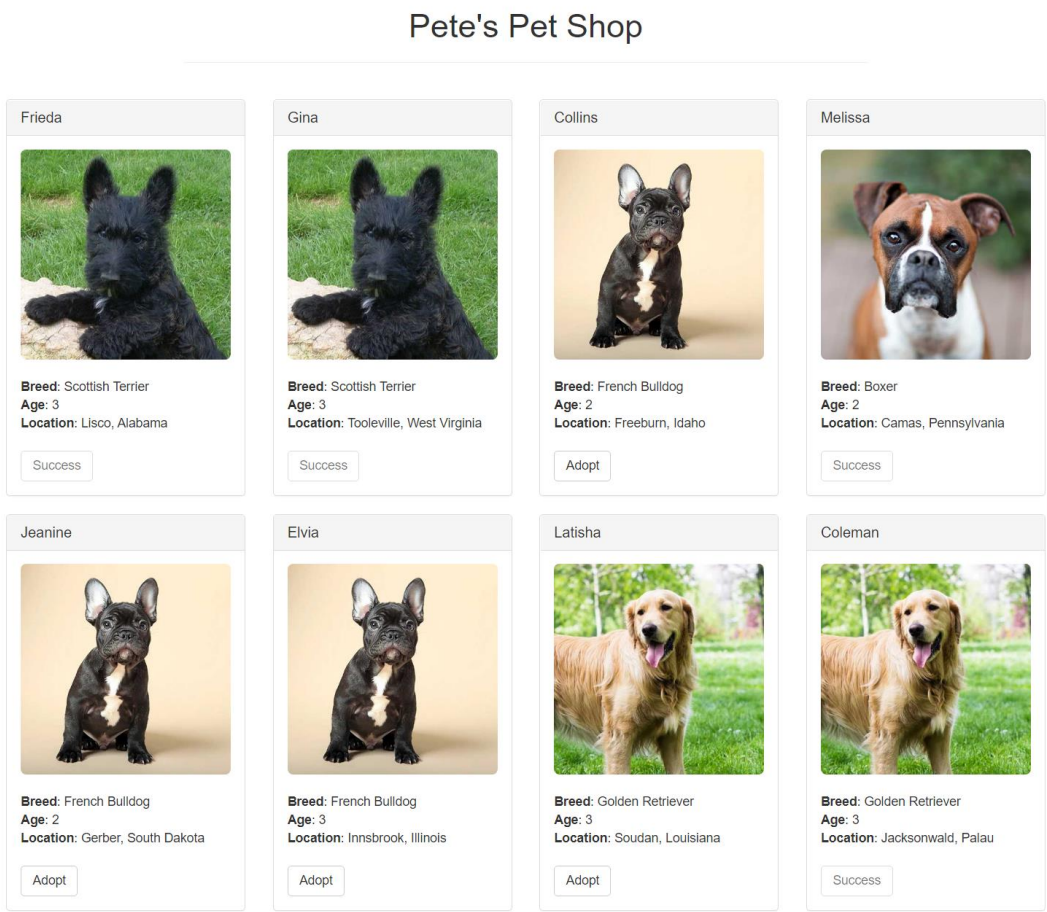
web3.js 적용하여 블록체인에 SmartContract 요청하기

[app.js] 수정

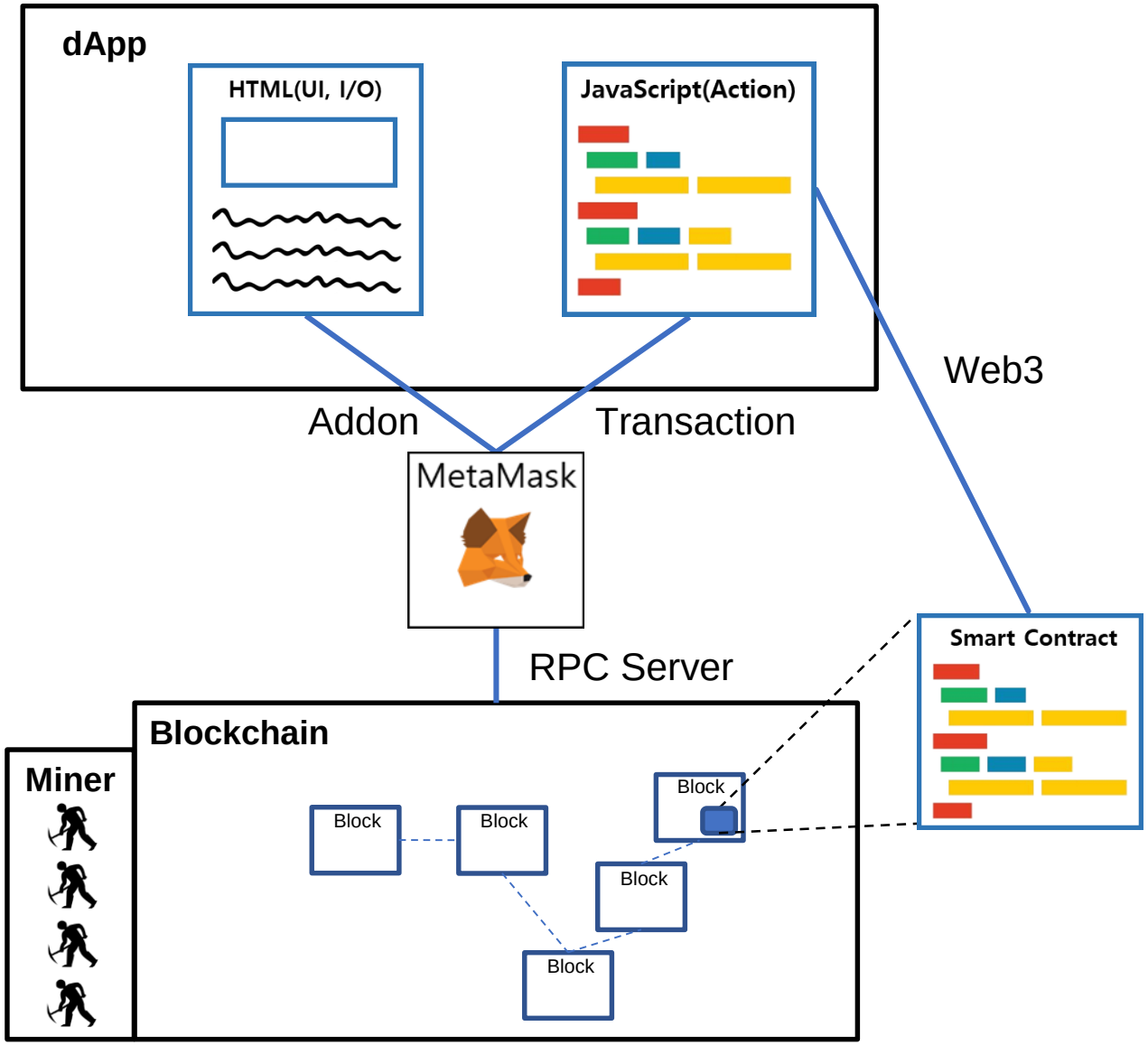
```
handleAdopt: function(event) {
  event.preventDefault();

  const petId = parseInt($(event.target).data('id'));

  let adoptionInstance;

  // metmask에 등록된 account 가져오기
  web3.eth.getAccounts(function(error, accounts) {
    if (error) {
      console.log(error);
    }


    const account = accounts[0];

    App.contracts.Adoption.deployed().then(function(instance) {
      adoptionInstance = instance;

      // adopt contract function 호출.
      return adoptionInstance.adopt(petId, {from: account});
    }).then(function(result) {
      // 성공시 markAdopted 호출
      return App.markAdopted();
    }).catch(function(err) {
      // 에러시 알림.
      alert(err.message);
    });
  });
}
```

04 Solidity 개발 맛보기 (pet-shop)

web3.js 적용하여 블록체인에 SmartContract 요청하기



04 Solidity 개발 맛보기 (pet-shop)



이더리움 dApp을 만드려면?

- HTML/CSS (보여지는 화면)
- javascript + Web3.js
- Solidity