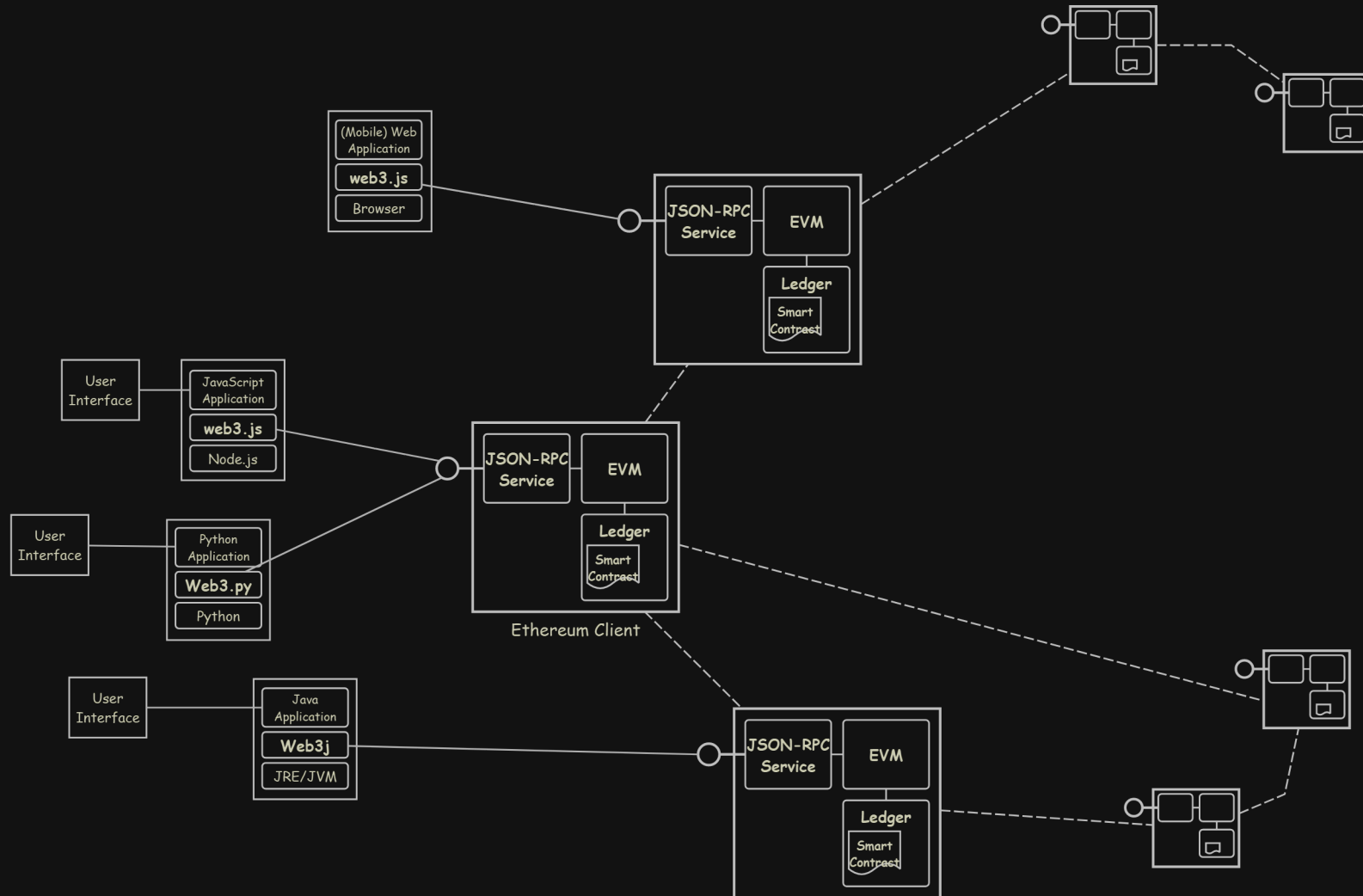




# DApp 개발 기초

오상문  
Feb. 2022







# DApp Architecture



# DApp 개발 Lifecycle

- 응용 분석/설계
- Smart Contract 구현
- Smart Contract **Audit**
- Smart Contract **Local** 배포/단위테스트
- Smart Contract **Testnet** 배포/단위테스트
- 응용 구현
- 응용 배포/단위테스트

# DApp 개발 환경/도구

Category		Tool/Service	Remarks
Editing		<u>Remix IDE</u>	<u>Web based</u>
Build/Deploy		<u>Truffle</u>	JavaScript based
		<u>Browine</u>	Python based
Local Client		<u>Ganache</u>	Ganache CLI
Mainnet/Testnet Gateway		<u>Infura</u>	
Library		<u>OpenZeppelin Contracts</u>	
Block Explorer		<u>Etherscan</u>	Mainnet
		<u>Etherscan/Rinkeby</u>	
Wallet		<u>MetaMask</u>	

# Truffle

## ■ Checking Node.js

```
$ node --version
```

## ■ Installing Truffle

```
$ npm ls -g truffle          # check whether or not Truffle is installed in global scope
$ npm uninstall -g truffle   # uninstalled currently installed Truffle
$ npm install -g truffle
```

## ■ Creating Truffle Project

```
$ mkdir smart-contract-101 && cd smart-contract-101
$ truffle init
...
$ ls
/  ../  contracts/  migrations/  test/  truffle-config.js
$ cat truffle-config.js | less
...
```

# Truffle Commands

```
$ truffle help
```

```
Truffle v5.4.22 - a development framework for Ethereum
```

```
Usage: truffle <command> [options]
```

```
Commands:
```

```
build      Execute build pipeline (if configuration present)
compile    Compile contract source files
...
console    Run a console with contract abstractions and commands available
...
deploy     (alias for migrate)
...
networks   Show addresses for deployed contracts on each network
...
test       Run JavaScript and Solidity tests
...
```

```
See more at http://trufflesuite.com/docs
```

```
$ truffle help migrate
```

```
...
```

```
$ truffle help test
```

```
...
```

- Resources

- [Truffle documentation](#)

- [Truffle commands](#)



# Truffle Configuration

## Prerequisite

- Setup **Node.js** project

```
$ npm init -y
$ npm install -D @truffle/hdwallet-provider
...
$ cat package.json
```

- Setup **Mnemonic**

- macOS

```
$ env | grep BIP39_MNEMONIC
...
$ echo 'export BIP39_MNEMONIC="..." ' >> ~/.profile
$ cat ~/.profile
...
```

- Windows

```
> setx BIP39_MNEMONIC="..."
```

- Setup **Infura Project**

- macOS

```
$ env | grep INFURA_PROJECT_ID
...
$ echo 'export INFURA_PROJECT_ID=...' >> ~/.profile
$ cat ~/.profile
...
```

- Windows

```
> setx INFURA_PROJECT_ID=...
```

- Resources

- Getting Started With Infura

# Truffle Configuration

## ■ truffle-config.js

```
const HDWalletProvider = require("@truffle/hdwallet-provider");
const mnemonic = process.env.BIP39_MNEMONIC;

module.exports = {
  networks: {
    development: {
      host: '127.0.0.1',
      port: 8545,
      network_id: 2016,
      gas: 3E8,
      gasPrice: 0,
      websockets: false
    },
    mainnet: {
      provider: () => new HDWalletProvider(
        mnemonic, "https://mainnet.infura.io/v3/" + process.env.INFURA_PROJECT_ID),
      network_id: '1'
    },
    rinkeby: {
      provider: () => new HDWalletProvider(
        mnemonic, "https://rinkeby.infura.io/v3/" + process.env.INFURA_PROJECT_ID),
      network_id: '4',
    }
  }
}
```

## ■ Resources

- [Truffle configuration reference](#)
- [@truffle/hdwallet-provider](#)
- [BIP-32 : Hierarchical Deterministic Wallets\(HD Wallets\)](#)
- [BIP-39 : Mnemonic code for generating deterministic keys](#)
- [Ethereum 201: HD Wallets](#)



# Testnets

Network(Chain)	Chain ID	Consensus	Avg. Block Time	Explorer
Mainnet	1	PoW	15 min.	<a href="#">Etherscan</a>
Ropsten	3	PoW	30 sec.	<a href="#">Etherscan/Ropsten</a>
<a href="#">Rinkeby</a>	4	PoA	15 sec.	<a href="#">Etherscan/Rinkeby</a>
<a href="#">Kovan</a>	42	PoA	4 sec.	<a href="#">Etherscan/Kovan</a>

- Resources
  - [Chainlist](#)
  - [EIP-155: Simple replay attack protection](#)

## Faucets

Network	Faucet
Ropsten	<a href="https://faucet.ropsten.be/">https://faucet.ropsten.be/</a>
Rinkeby	<a href="https://faucet.rinkeby.io/">https://faucet.rinkeby.io/</a>
	<a href="https://faucets.chain.link/rinkeby">https://faucets.chain.link/rinkeby</a>
Kovan	<a href="https://faucet.kovan.network/">https://faucet.kovan.network/</a>
	<a href="https://ethdrop.dev/">https://ethdrop.dev/</a>

# Truffle Console / Rinkeby

```
$ truffle console --network rinkeby
...
truffle(rinkeby)>
truffle(rinkeby)> web3.version
'1.5.3'
truffle(rinkeby)> web3.eth.net.getId()
4
truffle(rinkeby)> web3.eth.getBlockNumber().then(n => parseInt(n).toLocaleString())
'10,206,304'
truffle(rinkeby)> web3.eth.getBlock('latest')
...
truffle(rinkeby)> web3.eth.getBlock('latest').then(b => new Date(b.timestamp * 1000))
2022-02-21T10:46:03.000Z
truffle(rinkeby)> web3.eth.getBlock(0)
...
truffle(rinkeby)> web3.eth.getBlock(0).then(b => new Date(b.timestamp * 1000))
2017-04-12T14:59:06.000Z
truffle(rinkeby)> web3.eth.getCoinbase()
'0xb009cd53957c0d991cabe184e884258a1d7b77d9'
truffle(rinkeby)> web3.eth.getBalance(_).then(b => parseInt(b).toLocaleString())
'101,000,000,000,000,000'
truffle(rinkeby)> web3.eth.isMining()
false
truffle(rinkeby)> web3.eth.net.getPeerCount()
100
truffle(rinkeby)> web3.eth.getAccounts()
...
truffle(rinkeby)> accounts
...
truffle(rinkeby)> web3.eth.getBalance(accounts[0])
'101000000000000000'
truffle(rinkeby)> web3.eth.getBalance(accounts[1])
'100000000000000000'
truffle(rinkeby)>
```

# JSON-RPC and Web3.js


JSON-RPC	Description	web3.js
<code>eth_chainId</code>	the chain ID of the current connected node	<code>web3.eth.getChainId()</code>
<code>eth_blockNumber</code>	the number of most recent block	<code>web3.eth.getBlockNumber()</code>
<code>eth_getBlockByNumber</code>	information about a block by block number.	<code>web3.eth.getBlock()</code>
<code>web3_clientVersion</code>	the current client version	<code>web3.eth.getNodeInfo()</code>
<code>eth_coinbase</code>	the coinbase address to which mining rewards will go	<code>web3.eth.getCoinbase()</code>
<code>eth_accounts</code>	a list of addresses owned by client	<code>web3.eth.getAccounts()</code>
<code>eth_getBalance</code>	the balance of the account of given address	<code>web3.eth.getBalance()</code>
<code>eth_getTransactionCount</code>	the number of transactions sent from an address. (nonce)	<code>web3.eth.getTransactionCount()</code>
<code>eth_getCode</code>	code at a given address	<code>web3.eth.getCode()</code>
<code>eth_signTransaction</code>	signs a transaction can be submitted to the network at a later time.	<code>web3.eth.signTransaction()</code>
<code>eth_sendTransaction</code>	creates new message call transaction or a contract creation	<code>web3.eth.sendTransaction()</code>
<code>eth_sendRawTransaction</code>	creates new message call transaction or a contract creation for signed transactions.	<code>web3.eth.sendSignedTransaction()</code>
<code>eth_call</code>	executes a new message call immediately without creating a transaction on the block chain.	<code>web3.eth.call()</code>

## ■ Resources

- [JSON-RPC API](#)
- [Web3.js API](#)
- [Web3.py API](#)

# JSON-RPC Samples

```
$ curl -sSX POST --data '{"jsonrpc":"2.0","method":"eth_chainId","params":[],"id":21}' \
https://rinkeby.infura.io/v3/${INFURA_PROJECT_ID} | jq .
{
  "jsonrpc": "2.0",
  "id": 21,
  "result": "0x4"
}
$ curl -sSX POST --data '{"jsonrpc":"2.0","method":"eth_blockNumber","params":[],"id":22}' \
https://rinkeby.infura.io/v3/${INFURA_PROJECT_ID} | jq .
{
  "jsonrpc": "2.0",
  "id": 22,
  "result": "0x9bcd3"
}
$ curl -sSX POST --data '{"jsonrpc":"2.0","method":"eth_getBlockByNumber","params":["latest", false],"id":23}' \
https://rinkeby.infura.io/v3/${INFURA_PROJECT_ID} | jq .
...
$ curl -sSX POST --data '{"jsonrpc":"2.0","method":"eth_getBlockByNumber","params":["0x0", false],"id":24}' \
https://rinkeby.infura.io/v3/${INFURA_PROJECT_ID} | jq .
...
$ curl -sSX POST --data '{"jsonrpc":"2.0","method":"eth_accounts","params":[],"id":25}' \
https://rinkeby.infura.io/v3/${INFURA_PROJECT_ID} | jq .
{
  "jsonrpc": "2.0",
  "id": 25,
  "result": []
}
$
```

- Resources 
- `jq` : sed for JSON data

# Ganache (Ganache CLI)

Local standalone client mainly for testing

## ■ Installing

```
$ npm install -g ganache
...
$ ganache --help
```

## ■ Launching

```
$ ganache --chain.networkId 2016 \
--chain.chainId 2016 \
--server.host 127.0.0.1 \
--server.port 8545 \
--miner.defaultGasPrice 25000000000 \
--miner.defaultTransactionGasLimit 400000000 \
--miner.blockTime 0 \
--wallet.totalAccounts 15 \
--wallet.defaultBalance 10000 \
--wallet.unlockedAccounts 0 1 2 3 4 \
--database.dbPath run/ganache/data
```

## ■ Resources

### ■ Ganache startup options

## ■ Playing

```
$ truffle config get networks
{
  dashboard: {
    network_id: '*',
    networkCheckTimeout: 120000,
    url: 'http://localhost:24012/rpc',
    skipDryRun: true
  },
  development: {
    host: '127.0.0.1',
    port: 8545,
    network_id: 2016,
    gas: 300000000,
    gasPrice: 0,
    websockets: false
  },
  mainnet: { provider: [Function: provider], network_id: '1' },
  rinkeby: { provider: [Function: provider], network_id: '4' }
}
$ truffle console
truffle(development)>

...

truffle(development)> .exit
$
```

# Remix IDE

Best Solidity editor ever.

## ■ Intalling remixd

```
$ npm install @remix-project/remixd
...

$ npx remixd --help
```

## ■ Launching remixd

```
$ npx remixd --shared-folder ./ \
  --remix-ide https://remix.ethereum.org
...
$
```

## ■ Open <https://remix.ethereum.org>

- Click **Connect to Localhost** under the **File** section in the main pannel.
- In the **File Explorers**  on the left pannel, Click mouse right button on **contracts** directory, select **New Folder** menu to create **cryptopunks** directory under it.
- Click mouse right button on **contracts/cryptopunks** directory select **New Folder** menu to create **CryptoPunksMarket.sol** file under it.
- Copy the source from <https://github.com/Larvalabs/cryptopunks/blob/master/contracts/CryptoPunksMarket.sol> and paste it into the **contracts/cryptopunks/CryptoPunksMarket.sol** file.
- Change left pannel to **Solidity Compiler** by clicking  in the leftmost bar
- In the **Solidity Compiler**  on the left pannel, click **Compile** **CryptoPunksMarket.sol** button to compile the contract source.

# Sample Contract (1/4)

<https://github.com/larvalabs/cryptopunks/blob/master/contracts/CryptoPunksMarket.sol>

```
pragma solidity ^0.4.8;
contract CryptoPunksMarket {

    // You can use this hash to verify the image file containing all the punks
    string public imageHash = "ac39af4793119ee46bbff351d8cb6b5f23da60222126add4268e261199a2921b";

    address owner;

    string public standard = 'CryptoPunks';
    string public name;
    string public symbol;
    uint8 public decimals;
    uint256 public totalSupply;


    uint public nextPunkIndexToAssign = 0;

    bool public allPunksAssigned = false;
    uint public punksRemainingToAssign = 0;

    //mapping (address => uint) public addressToPunkIndex;
    mapping (uint => address) public punkIndexToAddress;

    /* This creates an array with all balances */
    mapping (address => uint256) public balanceOf;

    ...
}
```

- Resources 
  - Structure of Contract
    - State variables
    - Events
    - Functions
  - Data types
    - Boolean
    - Integer
    - Address
    - Fixed-sized byte array
    - Dynamically-sized byte array
    - String
    - Enum
    - Fixed-sized array
    - Dynamically-sided array
    - Mapping
  - Visibility



# Sample Contract (2/4)

<https://github.com/larvalabs/cryptopunks/blob/master/contracts/CryptoPunksMarket.sol>

```
...

struct Offer {
    bool isForSale;
    uint punkIndex;
    address seller;
    uint minValue;           // in ether
    address onlySellTo;      // specify to sell only to a specific person
}


struct Bid {
    bool hasBid;
    uint punkIndex;
    address bidder;
    uint value;
}

// A record of punks that are offered for sale at a specific minimum value, and perhaps to a specific
mapping (uint => Offer) public punksOfferedForSale;

// A record of the highest punk bid
mapping (uint => Bid) public punkBids;

mapping (address => uint) public pendingWithdrawals;

...
```

- Resources 
- [Structs](#)
- [Mapping Types](#)

# Sample Contract (3/4)


<https://github.com/larvalabs/cryptopunks/blob/master/contracts/CryptoPunksMarket.sol>

```
event Assign(address indexed to, uint256 punkIndex);
event Transfer(address indexed from, address indexed to, uint256 value);
event PunkTransfer(address indexed from, address indexed to, uint256 punkIndex);
event PunkOffered(uint indexed punkIndex, uint minValue, address indexed toAddress);
event PunkBidEntered(uint indexed punkIndex, uint value, address indexed fromAddress);
event PunkBidWithdrawn(uint indexed punkIndex, uint value, address indexed fromAddress);
event PunkBought(uint indexed punkIndex, uint value, address indexed fromAddress, address indexed toAddress);
event PunkNoLongerForSale(uint indexed punkIndex);

/* Initializes contract with initial supply tokens to the creator of the contract */
function CryptoPunksMarket() payable {
    // balanceOf[msg.sender] = initialSupply; // Give the creator all initial tokens
    owner = msg.sender;
    totalSupply = 10000; // Update total supply
    punksRemainingToAssign = totalSupply;
    name = "CRYPTOPUNKS"; // Set the name for display purposes
    symbol = "C"; // Set the symbol for display purposes
    decimals = 0; // Amount of decimals for display purposes
}

function setInitialOwner(address to, uint punkIndex) {
    if (msg.sender != owner) throw;
    if (allPunksAssigned) throw;
    if (punkIndex >= 10000) throw;
    ...
}

...
```

- Resources 
- Functions
  - View Functions
  - Pure Functions
- Events



# Solidity Features

## ✓ C, Java, JavaScript like syntax

Curly-brace block

## ✓ Statically typed

Compile-time type safety

## ✓ Imperative and object-oriented

Support `interface`, `abstract contract`, multiple `inheritance`

## ✓ Radically growing

Breaking change in every major version upgrade from `v0.5.0`(Nov 2018) to `v0.8.0`(Dec 2020)

## ✓ Runs on EVM<sup>Ethereum Virtual Machine</sup>

Compiled into bytecode and executed as a number of EVM opcodes.

### ■ Resources

- <https://github.com/ethereum/solidity>
- [Inheritance](#)
- [Interfaces](#)
- [Abstract Contracts](#)
- [EVM](#)
- [EVM Opcodes](#)
- [Ethereum Yellow Paper](#)

# Solidity Types (1/2)

Type	Keyword	Operators	Fields/Methods	Literal
Boolean	<code>bool</code>	<code>!</code> , <code>&amp;&amp;</code> , <code>  </code> , <code>==</code> , <code>!=</code>		
Unsigned Integer	<code>uint8</code> , <code>uint16</code> , <code>uint24</code> , ..., <code>uint248</code> , <code>uint256</code> , <code>uint</code>	<code>&lt;</code> , <code>&lt;=</code> , <code>==</code> , <code>&gt;=</code> , <code>&gt;</code> , <code>&amp;</code> , <code> </code> , <code>^</code> , <code>~</code> , <code>&lt;&lt;</code> , <code>&gt;&gt;</code> , <code>+</code> , <code>-</code> , <code>*</code> , <code>/</code> , <code>%</code> , <code>**</code>		<code>100</code> , <code>0x2eff</code> , <code>300_000_000</code> , <code>2e10</code> , <code>2.1e10</code>
Signed Integer	<code>int8</code> , <code>int16</code> , <code>int24</code> , ..., <code>int248</code> , <code>int256</code> , <code>int</code>	<code>&lt;</code> , <code>&lt;=</code> , <code>==</code> , <code>&gt;=</code> , <code>&gt;</code> , <code>&amp;</code> , <code> </code> , <code>^</code> , <code>~</code> , <code>&lt;&lt;</code> , <code>&gt;&gt;</code> , <code>+</code> , <code>-</code> , <code>*</code> , <code>/</code> , <code>%</code> , <code>**</code>		
Address	<code>address</code>		<code>balance</code> , <code>code</code> , <code>call()</code> , <code>delegatecall()</code> , <code>staticcall()</code>	<code>0xdCad3a6d3569DF655070DEd06cb7A1b2Ccd1D3AF</code>
Address Payable	<code>address payable</code>		<code>balance</code> , <code>code</code> , <code>call()</code> , <code>delegatecall()</code> , <code>staticcall()</code> , <code>transfer()</code> , <code>send()</code>	
Fixed-sized Byte Array	<code>byte1</code> , <code>byte2</code> , <code>byte3</code> , ... <code>byte31</code> , <code>byte32</code>	<code>&lt;</code> , <code>&lt;=</code> , <code>==</code> , <code>&gt;=</code> , <code>&gt;</code> , <code>&amp;</code> , <code> </code> , <code>^</code> , <code>~</code> , <code>&lt;&lt;</code> , <code>&gt;&gt;</code> , <code>x[k]</code>	<code>length</code>	

# Solidity Types (2/2)

Type	Keyword	Operators	Fields/Methods	Literal
(Dynamically-sized) Byte Array	<code>bytes</code>	<code>x[k]</code>	<code>push()</code> , <code>push(x)</code> , <code>pop()</code> , <code>bytes(string)</code> , <code>concat()</code>	
String	<code>string</code>		<code>concat()</code>	<code>'foo'</code> , <code>"foo"</code> , <code>'foo\nbar'</code> , <code>'foo\\bar'</code>
Array	<code>T[n]</code> , <code>T[]</code> , <code>T[n][m]</code> , <code>T[n][]</code> , <code>T[] [m]</code> , <code>T[][]</code>	<code>a[k]</code> , <code>a[m:n]</code>	<code>length</code> , <code>push()</code> , <code>push(x)</code> , <code>pop()</code>	[1, 2, 3]
Mapping	<code>mapping(key-type =&gt; value-type)</code>	<code>m[key]</code>		
Struct	<code>struct T { ... }</code>			

