블록 체인 : 모든 거래자의 전체 거래장부 공유 및 대조를 통해 거래를 안전하게 만드는 보안 기술

- 기존 거래 방식 : 거래내역을 작성하고 그 내용을 확인 후 거래가 일어남 (은행에서)
➔ 최소한만 저장 , 최소한의 인원만 접근하여 확인 (비밀은 최소한만 알고 보안하는게 안전하다고 생각했기 때문에 은행만 알고 있음)

- 블록체인 거래방식: 블록 이 존재(새로운 거래기록 저장),블록체인으로 연결된 pc 는 10 분 간격으로 거래내역 내용 비교 하고 과반수 이상 동의가 일어나면 블록화! ==> 인증받은 거래내역만 남게 된다 / 인증받지 못한 거래내역 폐기
거래자 거래장부 공유(암호화된)
➔ 따라서 거래내역 위조 어려움 (블록체인의 과반수 이상 다 해킹해서 거래장부를 속여야 하는데 불가능)

- 핵심 기술 : 해시 Hash (문장 길이에 관계없이 일정한 길이의 값으로 변경)
➔ 문장 내용이 완전히 같으면 동일한 완전히 같은 해시값 가짐! (하지만 문장이 조금이라도 다르면 완전히 다른 해시값 가짐)
➔ 해시값 조합을 통해 원문을 유추 할수 없음
➔ 적은 데이터양으로 원본내용 모두 완전히 같음을 비교 가능! (원본 내용이 엄청 많더라 하더라도 그 내용을 해시 값으로 변경해서 비교하면 적은 데이터양으로 비교 가능!)

암호화폐 : 블록체인에서 관리하고 있는 화폐(해쉬에 의해서)

장점: 기록물, 그 권한의 분산화(탈중앙화) ==> 의료, 금융, 물류 등등

1. 관리 효율성
2. 기록 신뢰, 보안성

이더리움

- 이더리움은 DApp 을 배포할 수 있는 탈중앙화 플랫폼이다.
- 현재 이더리움 엔진은 Go 언어와 C++, 파이썬 등으로 개발 되어있음
- 가장 활발하게 개발이 진행 되고 있는 것은 Go 로, go-ethereum(줄여서 Geth)
- 이더리움 엔진인 geth 은 3 가지 인터페이스를 통해 활용 가능(HTTP JSON RPC, web3.js, Solidity)

1) 설치(Geth : 이더리움의 전체 기능을 사용할 수 있는 풀 클라이언트)

```
$ brew tap Ethereum/Ethereum
$ brew install Ethereum
```

2) 하위 명령어 및 옵션

- 메인넷 네트워크 연결 : 이더리움 네트워크의 노드들은 기본적으로 30303 포트로 통신
(다른포트도 리스닝 가능)

2) go 설치(/usr/local/go)

- https://golang.org/dl/ 에서 osx10.8.pkg 설치

```
$ sudo vi .bash_profile ➔ export PATH=$PATH:/usr/local/go/bin (환경변수)
$ source ~/.bash_profile
$ go 또는 go env
```

hello 실습

```
$ vi hello.go
```

다음과 같이 작성해주세요.

```
/* hello.go - My first Golang program */
package main
import "fmt"
func main() {
    fmt.Printf("Hello, world\n")
}
```

자 이제 실행해봅시다.

```
$ go run hello.go
Hello, world
```