1. 스마트 계약 개요
   1) 블록체인에서 동작하는 응용 프로그램의 단위
   2) 튜링 완전한 고급 언어로 계약을 작성 => EVM 컴파일러로 컴파일해 EVM 바이트코드로 만듬
      =>블록체인에 배포
   3) 사용자는 브라우저나 콘솔에서 블록체인에 존재하는 EVM 바이트코드에 JSON-RPC 등으로 접근

2. 콘솔에서 계약 만들기
   1) 소스를 작성하고 컴파일을 하게 되면 ABI를 취득 * ABI란 계약의 외부 사양을 말함
   2) ABI로부터 계약 객체를 만듬
   3) 이 만들어진 계약 객체를 이더리움 블록체인에 배포
   * go-ethereum 1.6.0 이후 버젼부터 geth 상에서 RPC API 호출을 사용하여 solc를 호출하여 Solidity 소스를 컴파일 할 수 없음!(현재 버젼 : 1.8.4)

3. 계약 개발 환경
   1) Browser-Solidity(Remix)는 Solidity 언어 전용 웹 브라우저 기반 통합 개발 환경(IDE)로 다음과 같은 작업 수행이 가능
      a. 계약 코드 작성
      b. 컴파일
      c. 이더리움 노드에 배포
      d. 계약 매서드의 실행
   2) 실행 모드가 3가지 존재
      a. Javascript VM : geth 노드 연결 없이 모든 개발이 로컬 컴퓨터의 Remix의 메모리상에서 이루어짐
      b. injected Web3 : Mist나 MetaMask와 같이 Mist와 유사한 공급자에 의해 제공되는 실행 환경 이용
      c. Web3 provider : 로컬 컴퓨터에서 작동되는 geth 노드에 연결하여 수행, 이 환경에서는 트랜잭션이 네트워크를 통해 전달될 수 있음
   3) Remix에서 계약 만들기
      a. 터미널 상에 geth --identity "ChaChain" --rpc --rpcaddr “<내 아이피 주소>" --rpcport "8545" --rpccorsdomain "http://remix.ethereum.org" --datadir "/Users/chaminjun/data_testnet" --mine --minerthreads 1 --port "30303" --nodiscover --rpcapi "admin, debug, miner,shh,txpool,personal,db,eth,net,web3" --networkid 4649 console를 통해 구동
      b. http://remix.ethereum.org로 접근한다.(여기서 https가 아닌 http 프로토콜을 써야함)
      c. Run => Environment에서 Web3 provider 선택 후 http:내아이피주소:8545를 입력
      d. 콘솔창에 web3.personal.unlockAccount(web3.personal.listAccounts[0], “<password>”, 15000)를 입력하여 lock을 풀어준다.(mining이 진행중이여야 함!)
      e. Create 버튼을 누르게 되면 fullhash와 recipient가 나옴

4. 계약 개발
   1) Solidity 데이터 형식
      a. 정적으로 입력되는 언어이며, 컴파일 시 변수, 메서드의 인수와 반환값의 형태를 지정
      b. 값 형식과 참조형식이 있음
      c. 주소형식은 EOA나 계약 등의 계정 주소를 저장하며 크기는 20바이트이다. 또한 주소형식만의 메서드가 존재한다.
         - transfer : Ether를 송금하는데 실패 시 예외가 발생해 모든 처리를 없었던 것으로 돌려놓음
         - send : Ether를 송금하는데 실패해도 처리가 계속 진행되므로 반드시 되돌려 줄 값을 체크!
   2) 여러가지 실습은 직접 코드를 짜보면서 익힘
   3) 다음은 전역 변수로 취급된다.

a. block.blockhash(uint blockNumber) : 지정한 블록의 해시 값(bytes32)
b. block.coinbase : 해당 블록의 채굴자 주소(address)
c. block.number : 해당 블록의 번호(uint)
d. block.timestamp : 해당 블록의 타임스탬프(uint)
e. msg.sender : 송금자 주소, 현재의 호출처(address)
f. msg.value : 송금액(uint)

4) 필요 없어진 계약은 파기가 가능하고 파기하게 되면 해당 계약이 보유하고 있는 Ether는 지정한 주소로 송금된다.