

## 1. 계정 관련 명령

### 1) **web3.eth.personal.newAccount**

- a. 새로운 계정을 생성(안전하지 않은 웹소켓 또는 HTTP를 통해 이 함수를 호출하는 것은 좋지 않음)
- b. 인수로는 string형의 비밀번호이다.
- c. 반환 값은 string형의 생성된 계정 주소이다.

ex) `personal.newAccount("pass0")`

### 2) **web3.eth.accounts**

- a. 계정의 주소 목록을 표시한다. 배열 인덱스를 지정해 인덱스에 대응하는 계정 주소 표시
- b. 인수는 없다.
- c. 반환값은 array형으로 계정 주소들이다..

ex) `eth.accounts` 혹은 `eth.accounts[0]`

### 3) **web3.eth.coinbase**

- a. Etherbase 주소를 표시한다.
- b. 인수는 없다.
- c. 반환값은 string형으로 Etherbase 주소이다.

ex) `eth.coinbase`

### 4) **web3.miner.setEtherbase**

- a. Etherbase 주소를 설정한다.
- b. 인수로는 string형의 Etherbase로 설정할 계정 주소이다.
- c. 반환값은 boolean형으로 설정되었다면 true, 실패라면 예외를 처리한다.

ex) `miner.setEtherbase(eth.accounts[1])`

## 2. 채굴과 관련된 명령

### 1) **web3.eth.miner.start**

- a. 채굴 개시
- b. 인수로는 number형의 스레드 수량이다.
- c. 반환값은 boolean형으로 성공하였다면 true를 실패하면 예외를 처리한다.(null 반환)

ex) `miner.start(1)`

### 2) **web3.eth.miner.stop**

- a. 채굴 정지
- b. 인수는 없다.
- c. 반환값은 boolean형으로 성공하였다면 true를 실패하면 예외를 처리한다.

ex) `miner.stop()`

### 3) **web3.eth.mining**

- a. 채굴 중인지 확인
- b. 인수는 없다.
- c. 반환값은 boolean형으로 채굴중이라면 true를 아니라면 false를 반환한다.

ex) `eth.mining`

### 4) **web3.eth.hashrate**

- a. 현재 해시 속도를 표시
- b. 인수는 없다.
- c. 반환값은 number형으로 1초당 해시 속도이다.

ex) `eth.hashrate`

### 3. 블록과 관련된 명령

#### 1) **web3.eth.blockNumber**

- a. 현재 블록 번호를 표시한다.
- b. 인수는 없다.
- c. 반환값은 number형으로 블록번호이다.

ex) `eth.blockNumber`

#### 2) **web3.eth.getBlock**

- a. 블록 정보를 표시한다.
- b. 인수로는 블록번호가 들어간다.
- c. 반환값은 Object형으로 블록의 정보이다.

ex) `eth.getBlock(114322)`

### 4. 송금과 관련된 명령

#### 1) **web3.personal.unlockAccount**

- a. 지정한 계정의 잠금을 해제한다.
- b. 인수로는 총 3가지가 필요하다.
  - string형의 잠금 해제할 계정 주소
  - string형의 비밀번호
  - number형의 잠금 해제할 시간
- c. 반환값은 boolean형으로 성공하였다면 true를 실패하면 오류를 발생한다.

ex) `personal.unlockAccount(eth.accounts[0], "pass0", 0)`

#### 2) **web3.eth.sendTransaction**

- a. 지정한 주소로 송금한다. 송금이 완료되기 위해서는 채굴이 완료 즉 블록이 생성돼야 한다.
- b. 인수로는 Object형으로 송금자 주소, 송금처 주소, 송금액을 나타내는 객체이다.
- c. 반환값은 string형으로 트랜잭션 ID를 나타낸다.

ex) `eth.sendTransaction({from: eth.accounts[0], to: eth.accounts[1], value: web3.toWei(10, "ether")})`

#### 3) **eth.getBalance**

- a. 지정한 계정 주소의 잔고를 확인한다.
- b. 인수로는 string형의 계정 주소이다.
- c. 반환값은 number형으로 인수로 지정된 주소의 잔고이다.(단위는 wei)

ex) `eth.getBalance(eth.accounts[0])`

#### 4) **web3.fromWei**

- a. 1번째 인수 값을 2번째 단위로 변환
- b. 인수로는 총 2가지가 필요하다.
  - number형의 변환할 값
  - string형의 단위
- c. 반환값은 number형으로 2번째 단위값으로 변환된 1번째 인수 값이다.

ex) `web3.fromWei(eth.getBalance(eth.accounts[0]), "ether")`

#### 5) **web3.toWei**

- a. 지정한 값을 wei로 변환
- b. 인수로는 총 2가지가 필요하다.
  - number형의 변환할 값
  - string형의 단위
- c. 반환값은 number형으로 2번째 인수에서 지정한 단위의 첫 번째 값을 wei로 변환한 값

ex) `web.toWei(234, "ether")`

6) **web3.eth.getTransaction**

- a. 지정한 트랜잭션 ID의 정보를 표시
- b. 인수로는 string형의 트랜잭션 ID이다.
- c. 반환 값은 Object형으로 트랜잭션 정보이다.

ex) *eth.getTransaction*("트랜잭션 ID")

7) **web3.eth.pendingTransactions**

- a. 대기 중인 트랜잭션 정보를 표시
- b. 인수는 없다.
- c. 반환값으로는 Array형으로 대기 중인 트랜잭션 정보를 표시한다.

ex) *eth.pendingTransactions*