Chapter 01. Node 알아보기

* 동기 / 비동기 입출력
* 동기 입출력 (Blocking I/O) : 순차적인 기능 실행. 한 가지 작업이 끝난 이후에 다음 작업이 시작된다
* 비동기 입출력(Non-Blocking I/O) : 기능 실행에 대기가 없이 진행. 콜백 함수(Callback Function)을 통해, 작업 완료 시 불러올 함수를 설정할 수 있다.
* Callback Function : javascript에서는 변수에 함수를 할당할 수 있으며, 다른 함수의 내부에서 파라미터로 전달된 함수를 호출하는 것을 의미한다.
* Blocking I/O

**var contents = file.read(‘a.txt’);**

(대기)

doShow(contents);

var result = doAdd(10, 10);

※ Non-Blocking I/O

**file.read(‘a.txt’, function(contents){**

**doShow(contents);**

});

var result = doAdd(10,10);

* doAdd(10, 10)은 file.read가 시작되고 바로 동시에 실행되며, fire.read가 완료되면 파일 시스템에서 콜백 함수를 호출한다.
* 이벤트 기반 입출력
* 노드는 Event Driven, Non-blocking I/O 모델을 사용한다.
* Event Binding : 버튼을 클릭하면 🡪 click 이벤트 발생  
   🡪 이 click 이벤트를 함수 객체와 ‘binding’ 🡪 click 이벤트 발생 시   
   등록한 함수 객체 실행
* http로 웹 서버로 request 보내고 response를 받는 예시

http.request(options, function(res){

res.on(‘data’, function(chunk){

console.log(‘BODY ; ‘ + chunk);

});

};

);

* 서버에 request를 보낸 후 response를 받으면 function(res)…를 실행
* res.on(‘data’, function( ){ }); 에서 data란 이벤트가 발생하면 이와 바인딩 된 function(chunk)… 함수가 실행되는 구조. 응답으로 받은 데이터는 chunk에 담긴다.