

다음은 promise를 사용하는 간단한 예제이며, 각 상태가 어떻게 연결되는지 보여주고 있습니다.

```
var test = function(bool){  
  return new Promise(function(resolve, reject){  
    setTimeout(function(){  
      if(bool){  
        resolve("fulfilled 상태입니다. then으로 연결됩니다.");  
      }  
      else{  
        reject("rejected 상태입니다. catch로 연결됩니다.");  
      }  
    }, 1000)  
  })  
}  
  
test(true)  
  .then(function(result){  
    console.log(result);  
  })  
  .catch(function(err){  
    console.log(err)  
  })
```

먼저 Promise 객체를 생성하는 부분입니다.

Promise 생성자 함수에 콜백함수를 정의하는데요, 매개변수로 resolve, reject를 정의합니다.

처리하는 로직상에서 비동기 처리가 정상적으로 수행되면 resolve() 함수를 호출하고, 비동기 처리가 실패하면 reject() 함수를 호출합니다.

resolve 함수가 호출되면,  
'then()' 함수가  
이어서 호출됨.

reject 함수가 호출되면,  
'catch()' 함수가  
이어서 호출됨.