

JavaScript

≡ 다중 선택

JavaScript

promise

```
function add10(num) {
  const promise = new Promise((resolve, reject) => {
    // 비동기 작업 실행하는 함수
    // executor
    setTimeout(() => {
      if (typeof num === "number") {
        resolve(num + 10);
      } else {
        reject("num이 숫자가 아닙니다.");
      }
    }, 2000);
  });

  return promise;
}

add10(0)
// then 메서드: promise의 비동기 작업이 성공했을 경우 실행
.then((result) => {
  console.log(result);
  return add10(result);
})
.then((result) => {
  console.log(result);
  return add10(result);
})
// catch 메서드 : promise의 비동기 작업이 실패했을 경우 실행
.catch((error) => {
```

```
    console.log(error);  
  });
```

async & await

```
// async  
// 어떤 함수를 비동기 함수로 만들어주는 키워드  
// 함수가 프로미스를 반환하도록 변환해주는 그런 키워드  
  
async function getData() {  
  return new Promise((resolve, reject) => {  
    setTimeout(() => {  
      resolve({  
        name: "이정환",  
        id: "winterlood",  
      });  
    }, 1500);  
  });  
}  
  
// await  
// async 함수 내부에서만 사용이 가능한 키워드  
// 비동기 함수가 다 처리되기를 기다리는 역할  
// 이것 사용하면 then 메서드 사용할 필요 없음  
  
async function printData() {  
  const data = await getData();  
  console.log(data);  
}
```