

# 비동기

동시에 발생하지 않음 -> 순차적으로 발생

c.f. 동기 : 동시에 발생함

동기 :

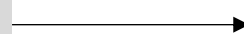
작업 A

데이터 a가 필요한 작업 B

아직 작업 A가 진행 중이므로 데이터 a가 존재하지 않음 (에러)

비동기 :

작업 A



데이터 a가 필요한 작업 B

Chapter 03.

웹 프론트 언어로의 JS

- 07. 비동기

# 비동기

js\_basic\_practice4

Chapter 03.  
웹 프론트 언어로의 JS  
- 07. 비동기

1. 콜백 함수 : 다른 함수의 인자로 넘겨져서 호출되는 함수

```
function A (callback) {                A(B(C(D(E()))));  
    // 수행해야 할 코드  
    callback(a);                        // A의 코드 흐름을 보려면 E 함수까지 타고 들어가야 함  
}
```

```
function B (a) {  
    // 데이터 a를 이용한 코드  
}
```

```
A(B);
```

# 비동기

js\_basic\_practice4

Chapter 03.  
웹 프론트 언어로의 JS  
- 07. 비동기

## 2. 프라미스 (Promise)

```
const promise = new Promise(function(resolve, reject) {  
  // 코드 구현  
  resolve(value); // 성공 시 resolve 호출  
  reject(error); // 실패 시 reject 호출  
}  
  
promise.then((result) => {  
  // 성공 시 넘어오는 값 result로 코드 구현  
}).catch((error) => {  
  // 실패 시 넘어오는 에러 error로 코드 구현  
});
```

```
A()  
  .then((result) => {  
    B(result);  
  })  
  .then((result) => {  
  })  
  ...
```

# 비동기

js\_basic\_practice4

Chapter 03.  
웹 프론트 언어로의 JS  
- 07. 비동기

## 3. async와 await

```
const A = async () => {  
  // 코드  
  return a;  
};  
const B = (a) => {};
```

```
const a = await A();  
B(a);
```

```
const a = await A();  
const b = await B(a);  
const c = await C(b);  
const d = await C(c);  
const e = await C(d);
```