

JavaScript (ES6) For Framework

13. callback



함수의 실행 순서

```
function a() {  
    console.log("a 실행");  
}
```

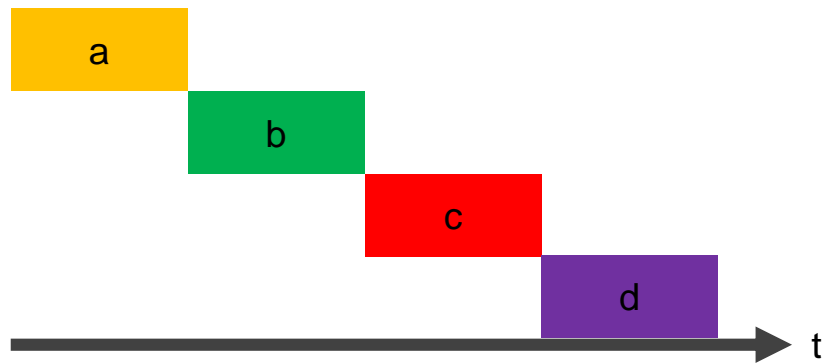
```
function b() {  
    console.log("b 실행");  
}
```

```
function c() {  
    console.log("c 실행");  
}
```

```
function d() {  
    console.log("d 실행");  
}
```

```
a();  
b();  
c();  
d();
```

동기와 비동기



동기와 비동기

```
function a() {  
  console.log("a 실행");  
}
```

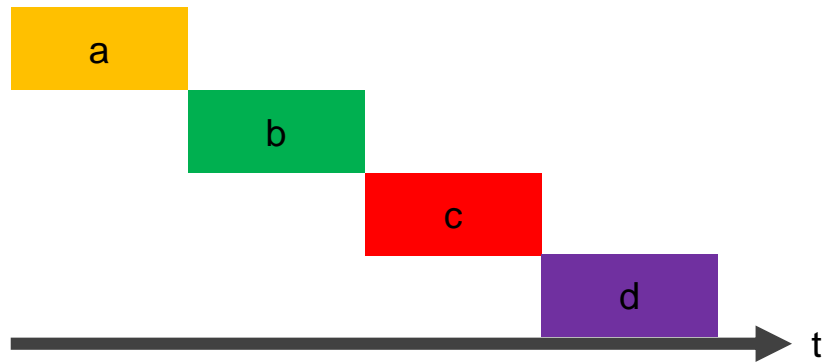
```
function b() {  
  console.log("b 실행");  
}
```

```
function c() {  
  console.log("c 실행");  
}
```

```
function d() {  
  console.log("d 실행");  
}
```

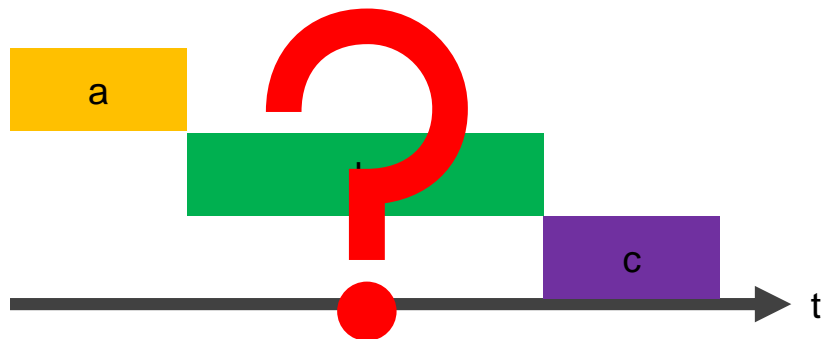
```
a();  
b();  
c();  
d();
```

← 실행에 3초 걸린다면?



동기와 비동기

```
function a() {  
  console.log("a 실행");  
}  
  
function b() {  
  console.log('오래 걸리는 b 실행');  
}  
  
function c() {  
  console.log("c 실행");  
}  
  
a();  
setTimeout(b, 3000);  
c();
```



동기와 비동기

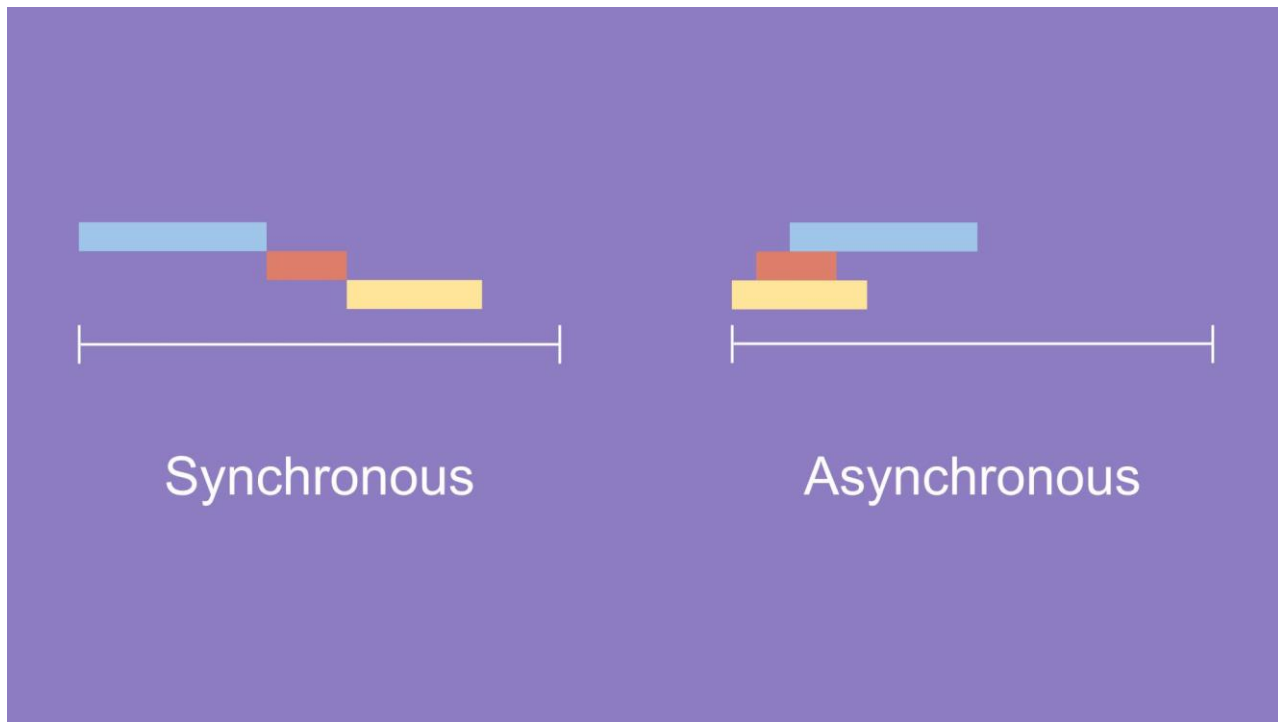
```
function a() {  
  console.log("a 실행");  
}  
  
function b() {  
  console.log('오래 걸리는 b 실행');  
}  
  
function c() {  
  console.log("c 실행");  
}  
  
a();  
setTimeout(b, 3000);  
c();
```



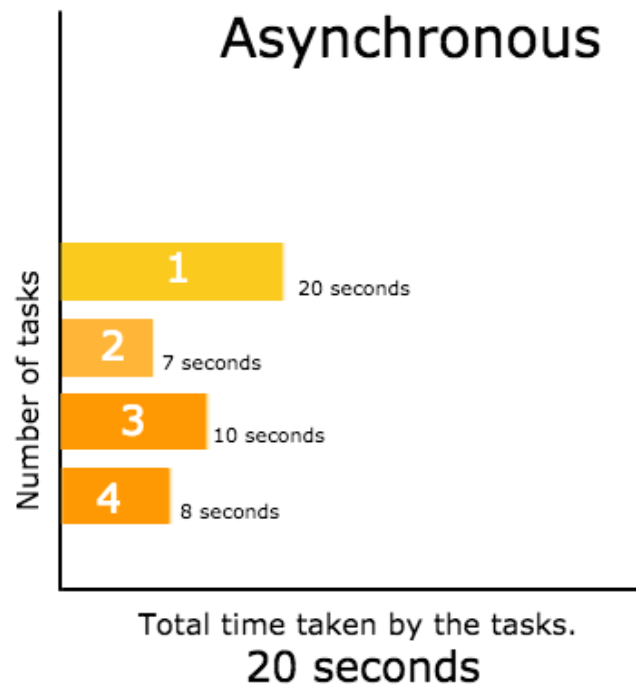
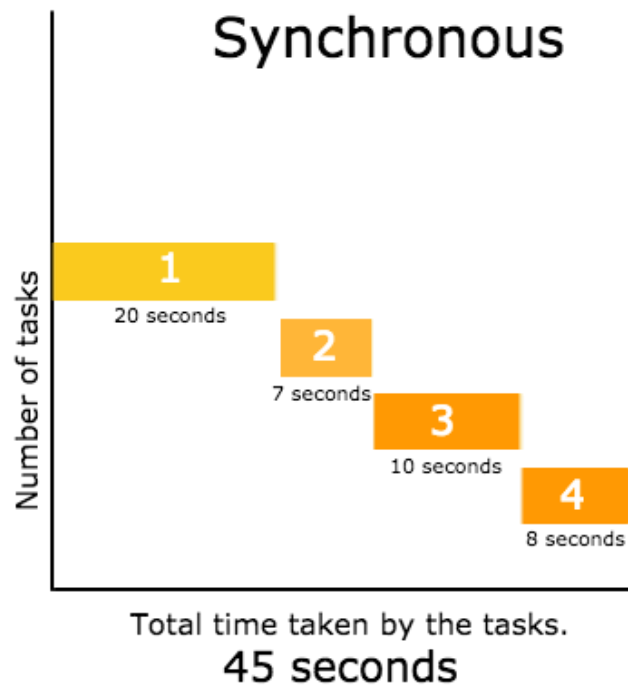
- 바로 답을 내놓을 수 없는 상황이 아니라면, 천년만년 기다리고 있지 않는다.
- 실행 순서 보장 안된다.

=비동기처리

동기와 비동기



동기와 비동기



“

Web 상에서의 비동기처리

“

Web 상에서의 비동기처리

Web에서의 동기적인 처리는 비효율적...

“

그럼에도 불구하고 **실행순서가 보장되어야** 하는 경우는 있다

결제처리 함수가 완료되면 → B함수 실행된다
오래 걸리는 C 함수의 리턴 값으로 → D함수가 처리된다

Callback 함수

callback 함수

Callback 함수

callback 함수

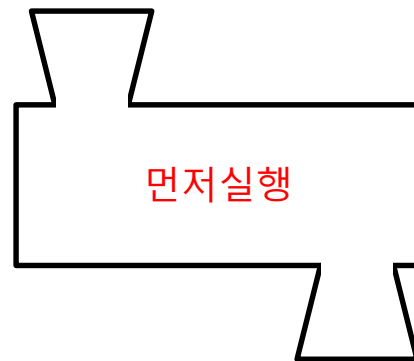
Callback 함수

좀 이따 부르면 그 때 오는 함수

Callback 함수

부르면 타이밍 맞춰 오는 함수

Callback 함수

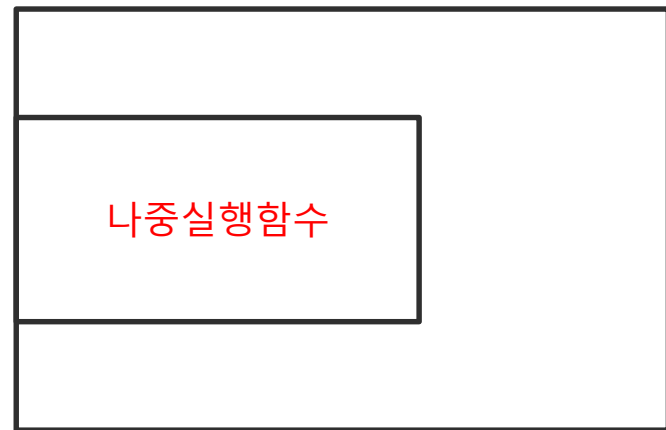


Callback 함수

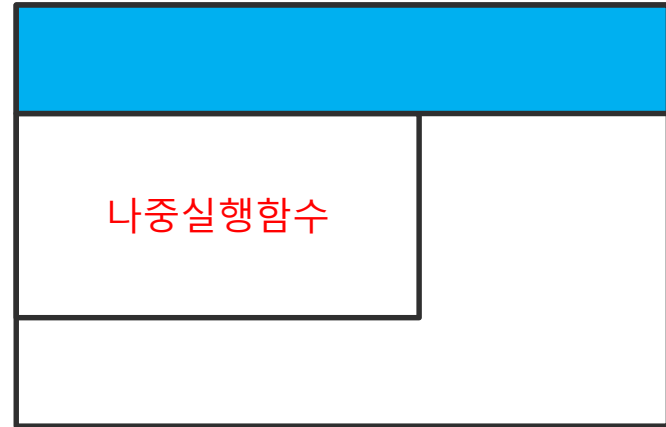
나중실행함수

먼저실행함수

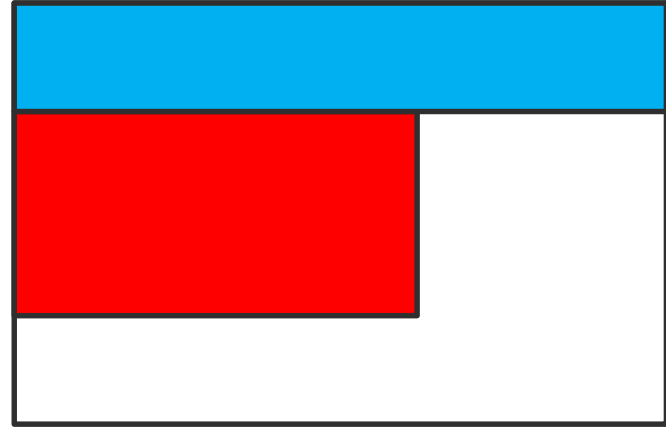
Callback 함수



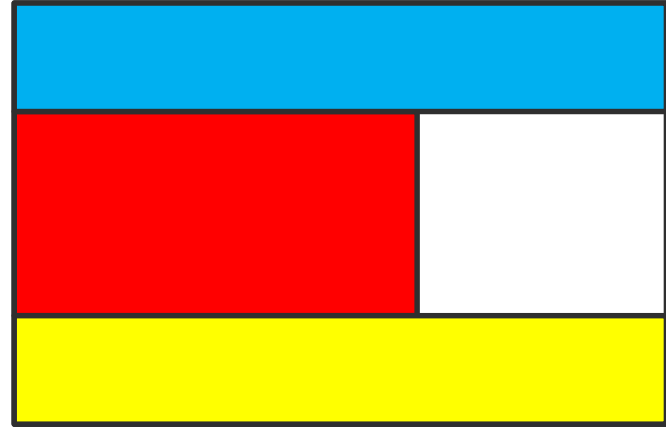
Callback 함수



Callback 함수



Callback 함수



콜백 지옥

5초 걸리는 a,

4초 걸리는 b,

3초 걸리는 c,

2초 걸리는 d가

순차적으로 실행되어야 한다면?

콜백 지옥

```
function a() {  
  setTimeout(function b() {  
    console.log('5초 걸리는 a');  
    setTimeout(function c() {  
      console.log('4초 걸리는 b');  
      setTimeout(function d() {  
        console.log('3초 걸리는 c');  
        setTimeout(function e() {  
          console.log('2초 걸리는 d');  
        }, 2000);  
      }, 3000);  
    }, 4000);  
  }, 5000);  
};  
  
a();
```

Callback 지옥

ES6: Promise~async

다음강의

Section 14

let & const

Lecturer Github : <https://github.com/kangtegong/>

수업자료링크 : <https://github.com/kangtegong/develop-javascript>

Email : tegongkang@gmail.com