

# 1~2. 리액트 학습 준비 (실습3)

Prof. Seunghyun Park ([sp@hansung.ac.kr](mailto:sp@hansung.ac.kr))

Mobility and Cybersecurity Lab. <https://sp-mcslab.github.io>

Division of Computer Engineering

## 학습 목표: 2. 리액트를 위한 자바스크립트

---

- ES2015+
  - 변수와 상수
  - 함수
  - 화살표 함수
  - 객체와 this
  - script type 확인
- 비동기 자바스크립트 구현

# 실습7: 비동기 자바스크립트 구현

예제) 3개의 텍스트 파일을 순서대로 읽어와 콘솔에 출력하는 프로그램 구현하기

```
start
1st reading: This file contains sample text #1.
2nd reading: This file contains sample text #2.
3rd reading: This file contains sample text #3.
end
```



readme1.txt



readme2.txt



readme3.txt

```
/* ch02/ex7/00-readfile.js */
const fs = require('fs');

fs.readFile('./readme1.txt', (err, data) => {
  if (err) {
    console.error(err);
  }
  else {
    console.log('1st reading:', data.toString());
  }
});
```

1st reading: This file contains sample text #1.

# 실습7: 비동기 자바스크립트 구현 (1)

- 문제1: 3개의 비동기 함수를 연속해서 호출

```
// code1
console.log('start');
fs.readFile(filename1, callback1);
fs.readFile(filename2, callback2);
fs.readFile(filename3, callback3);
console.log('end');
```

```
start
end
3rd reading: This file contains sample text #3.
1st reading: This file contains sample text #1.
2nd reading: This file contains sample text #2.
```

```
start
end
1st reading: This file contains sample text #1.
3rd reading: This file contains sample text #3.
2nd reading: This file contains sample text #2.
```

## 실습7: 비동기 자바스크립트 구현 (2)

- 문제2: callback 함수 내부에서 비동기 함수를 호출

```
// code2
console.log('start');
fs.readFile(filename1, () => { callback1
  console.log(data1);
  fs.readFile(filename2, () => { callback2
    console.log(data2);
    fs.readFile(filename3, () => { callback3
      console.log(data3);
      console.log('end');
    });
  });
});
```

start

1st reading: This file contains sample text #1.

2nd reading: This file contains sample text #2.

3rd reading: This file contains sample text #3.

end

# 실습7: 비동기 자바스크립트 구현 (3)

- 문제3: Promise 객체 활용

```
// code3
const promise = new Promise((resolve, reject) => {
  console.log('start');
  fs.readFile(filename1, (err, data1) => {
    if (!err) resolve(data1);
    else reject(err); // callback1
  })
});

promise
  .then(data1 => {
    console.log(data1);
    return new Promise((resolve, reject) => {
      fs.readFile(filename2, callback2)
    })
  })
  .then(data2 => {
    console.log(data2);
    return new Promise((resolve, reject) => {
      fs.readFile(filename3, callback3)
    })
  })
  .then(data3 => console.log(data3))
  .catch(err => console.error())
  .finally(() => console.log('end'))
```

# 실습7: 비동기 자바스크립트 구현 (4)

- 문제4: fs.promises 모듈 활용

```
// code4
console.log('start');
fsPromises.readFile(filename1)
  .then(data1 => {
    console.log(data1);
    return fsPromises.readFile(filename2)
  })
  .then(data2 => {
    console.log(data2);
    return fsPromises.readFile(filename3);
  })
  .then(data3 => console.log(data3))
  .catch(err => console.error())
  .finally(() => console.log('end'))
```

# 실습7: 비동기 자바스크립트 구현 (5)

- 문제5: async/await 구문 활용

```
// code5
console.log('start');
(async () => {
  try {
    let data1 = await fsPromises.readFile(filename1);
    console.log(data1);
    let data2 = await fsPromises.readFile(filename2);
    console.log(data2);
    let data3 = await fsPromises.readFile(filename3);
    console.log(data3);
  }
  catch(err){
    console.error();
  }
  finally{
    console.log('end');
  }
})();
```