

9기 자기주도 PJT

비동기 프로그래밍



자기주도 PRJ : 비동기 프로그래밍

최 인 국 Project consultant

- Cymer(2006, 10 ~) 반도체 장비 모니터링 시스템 개발
- Future System(2013, 05 ~) 네트워크 보안 장비 개발
- ESTMOB(2014, 06 ~) 파일 전송 솔루션
- 프리랜서 (2017, 03 ~) 카메라 제어, 출퇴근 시스템 등 개발

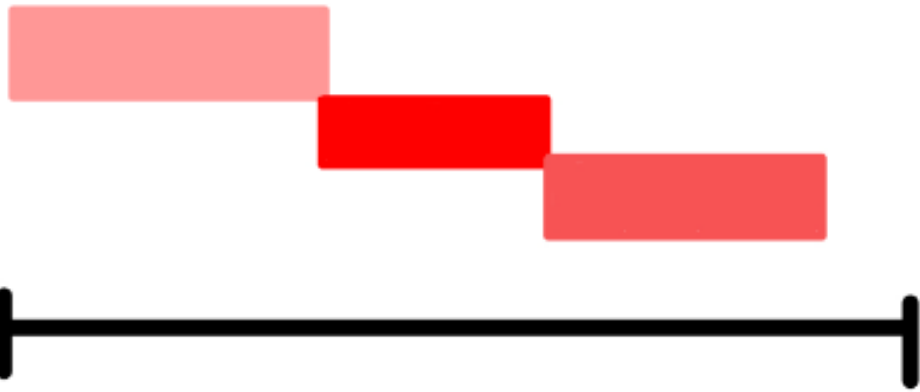
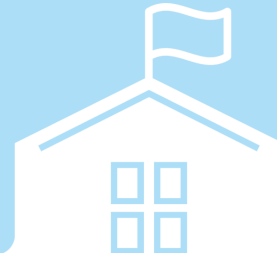


동기 vs 비동기



- ✓ 동기 - Task 순차 처리
- ✓ 비동기 - Task 병렬 처리

동기 vs 비동기



Synchronous
동기



Asynchronous
비동기

비동기 함수



```
function delay_word(word, delay) {  
  return new Promise(resolve => {  
    setTimeout(function () {  
      resolve(word)  
    }, delay)  
  })  
}
```

Chaining 처리, Hard code



```
delay_word('SAMSUNG', 500).then((resolve) => {  
  
  console.log(resolve)  
  
  delay_word('SW', 490).then((resolve) => {  
  
    console.log(resolve)  
  
    delay_word('ACADEMY', 480).then((resolve) => {  
  
      console.log(resolve)  
  
      delay_word('FOR', 470).then((resolve) => {  
  
        console.log(resolve)  
  
        delay_word('YOUTH', 460).then((resolve) => {  
  
          console.log(resolve)  
  
        })  
      })  
    })  
  })  
})
```

```
async function test(){  
  const resolve_0 = await delay_word('SAMSUNG', 500)  
  console.log(resolve_0)  
  const resolve_1 = await delay_word('SW', 490)  
  console.log(resolve_1)  
  const resolve_2 = await delay_word('ACADEMY', 480)  
  console.log(resolve_2)  
  const resolve_3 = await delay_word('FOR', 470)  
  console.log(resolve_3)  
  const resolve_4 = await delay_word('YOUTH', 460)  
  console.log(resolve_4)  
}
```

Chaining 처리, Soft code



```
const array = [{word:'SAMSUNG', delay:500}, {word:'SW', delay:490},  
               {word:'ACADEMY', delay:480}, {word:'FOR', delay:470},  
               {word:'YOUTH', delay:460}]  
  
array.reduce((prev, item) => {  
  return prev.then(() =>  
    delay_word(item.word, item.delay).then((promise) => {console.log(promise)}))  
}, Promise.resolve())
```

```
const array = [{word:'SAMSUNG', delay:500}, {word:'SW', delay:490},  
               {word:'ACADEMY', delay:480}, {word:'FOR', delay:470},  
               {word:'YOUTH', delay:460}]  
  
async function test(){  
  for(const item of array) {  
    const resolve = await delay_word(item.word, item.delay)  
    console.log(resolve)  
  }  
}
```

All 처리, 비 순차 결과



```
const array = [{word:'SAMSUNG', delay:500}, {word:'SW', delay:490},  
               {word:'ACADEMY', delay:480}, {word:'FOR', delay:470},  
               {word:'YOUTH', delay:460}]  
  
array.forEach(async (item) => {  
  
    delay_word(item.word, item.delay).then((resolve) => {console.log(resolve)})  
})
```

```
const array = [{word:'SAMSUNG', delay:500}, {word:'SW', delay:490},  
               {word:'ACADEMY', delay:480}, {word:'FOR', delay:470},  
               {word:'YOUTH', delay:460}]  
  
array.forEach(async (item) => {  
  
    const resolve = await delay_word(item.word, item.delay)  
  
    console.log(resolve)  
  
})
```


All 처리, 순차 결과



```
const array = [{word: 'SAMSUNG', delay: 500}, {word: 'SW', delay: 490},  
               {word: 'ACADEMY', delay: 480}, {word: 'FOR', delay: 470},  
               {word: 'YOUTH', delay: 460}]  
  
const promise_list = []  
array.forEach((item) => {  
    const promise = delay_word(item.word, item.delay)  
    promise_list.push(promise)  
})  
  
Promise.all(promise_list).then((values) => {  
    values.forEach((resolve) => {console.log(resolve)})  
})
```

```
const array = [{word: 'SAMSUNG', delay: 500}, {word: 'SW', delay: 490},  
               {word: 'ACADEMY', delay: 480}, {word: 'FOR', delay: 470},  
               {word: 'YOUTH', delay: 460}]  
  
async function test(){  
    const async_fun_list = []  
  
    for(item of array){  
        const async_fun = delay_word(item.word, item.delay)  
        async_fun_list.push(async_fun)  
    }  
  
    for(async_fun of async_fun_list){  
        const resolve = await async_fun  
        console.log(resolve)  
    }  
}
```



과제

Source



<https://lab.ssafy.com/crazygun22/shared/-/tree/master/async/test>

비동기 함수

```
const axios = require('axios');

async function request(sub_path){

    const url = 'http://13.124.193.201:8844/' + sub_path

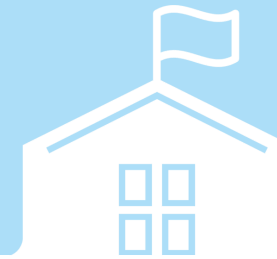
    try{

        const response = await axios.get(url);

        return response.data
    }
    catch(e){

        console.log(e)
    }
}
```

비동기 함수



Chaining 처리, Hard Code	<code>promise_hard_code.js</code> (제공됨) <code>await_hard_code.js</code>
Chaining 처리, Soft Code	<code>promise_soft_code.js</code> <code>await_soft_code.js</code>
All 처리, 비 순차 결과	<code>promise_all_non_sequence.js</code> <code>await_all_non_sequence.js</code>
All 처리, 순차 결과	<code>promise_all_sequence.js</code> <code>await_all_sequence.js</code>



Q/A

여러분의
폭풍 성장을 응원합니다!!

