

1번 실습 (Async, Await)

Async Await이란?

Async Await 문법은 가장 최근에 나온 비동기 처리 문법중 하나로 기존의 Promise의 단점을 보완하고자 만들어졌습니다. Async Await을 사용하면 가독성이 좋은 비동기 처리 코드를 작성하실 수 있을 겁니다.

function 앞에 `async` 를 붙이면 해당 함수는 항상 프라미스를 반환합니다. 프라미스가 아닌 값을 반환하더라도 이행 상태의 프라미스(resolved promise)로 값을 감싸 이행된 프라미스가 반환되도록 합니다.

`async` 가 붙은 함수는 반드시 프라미스를 반환하고, 프라미스가 아닌 것은 프라미스로 감싸 반환합니다. 굉장히 간단하죠? 그런데 `async` 가 제공하는 기능은 이뿐만이 아닙니다. 또 다른 키워드 `await` 는 `async` 함수 안에서만 동작합니다.

자바스크립트는 `await` 키워드를 만나면 프라미스가 처리될 때까지 기다립니다. 결과는 그 이후 반환됩니다.

Syntax

```
async function 함수이름(매개변수) {
  //코드
  //(await을 써야 한다면)
  await 비동기 처리 함수
}
```

예시 1

```
async function get1() {
  return 1;
}

get1().then(alert); // 1
```

예시 2

```
function after2Seconds() {
  return new Promise((resolve, reject) => {
    setTimeout(() => {
      resolve("Resolved");
    }, 2000);
  });
}

async function asyncCall() {
  console.log('calling');
  const result = await after2Seconds();
  console.log(result);
}

asyncCall();
```

기억해야 할 점

- 프로미스를 사용하는 친구다

- 함수 앞에 async를 적어줘야 await를 쓸 수 있다
- await은 promise에서 리턴값이 반환되기 전까지 기다린다
- async 함수는 항상 프로미스를 반환한다

▼ 실습 정답 코드

```
//async, await으로 변환하는 방법

// 1. Promise 대신 async로 비동기 처리해 'elice'를 반환하도록 fetchUser 함수를 수정하세요.
async function fetchUser() {
  return "elice";
}

const user = fetchUser();
user.then(console.log);

// 2. delay 함수를 이용해 getCoffee와 getTea 함수를 작성하세요.
function delay(ms) {
  return new Promise((resolve) => setTimeout(resolve, ms));
}

async function getCoffee() {
  await delay(1000);
  return "coffee";
}

async function getTea() {
  await delay(1000);
  return "tea";
}

// 3. 위 두 개의 함수를 사용해서 coffee와 tea를 한번에 반환합니다.
async function getDrinks() {
  const coffee = await getCoffee();
  const tea = await getTea();
  return `${coffee} and ${tea}`;
}

getDrinks().then(console.log);
```

2번 실습

힌트

```
const btn = document.getElementById("btn");
const mes = document.getElementById("message");

async function triathlon() {
  let swimEnd = await exercise(0, "swim");
  let bicycleEnd = await exercise(swimEnd, "bicycle");
  let runEnd = await exercise(bicycleEnd, "run");

  return runEnd;
}

const delay = (ms) => new Promise((resolve) => setTimeout(resolve, ms));

async function exercise(startTime, name) {
  const exerciseInput = document.getElementById(name);
  // 1. 완주에 걸리는 시간을 구하세요.
  // parseInt를 이용하여 exerciseInput.value를 변수에 저장하세요
```

```
// 2. setTimeout 대신 delay 함수를 사용해 비동기처리를 하세요.
// delay함수를 이용해보세요.
// mes에 innerHTML을 이용하여 메시지를 적어넣어보세요

// 3. 완주한 후의 시간을 반환하세요.

return 0;
}

btn.addEventListener("click", () => {
  triathlon().then((param) => {
    mes.innerText += `total time : ${param}`;
  });
});
```

▼ 실습 정답 코드

```
const btn = document.getElementById("btn");
const mes = document.getElementById("message");

async function triathlon() {
  let swimEnd = await exercise(0, "swim");
  let bicycleEnd = await exercise(swimEnd, "bicycle");

  let runEnd = await exercise(bicycleEnd, "run");

  return runEnd;
}

const delay = (ms) => new Promise((resolve) => setTimeout(resolve, ms));

async function exercise(startTime, name) {
  const exerciseInput = document.getElementById(name);
  // 1. 완주에 걸리는 시간을 구하세요.
  const t = parseInt(exerciseInput.value);

  // 2. setTimeout 대신 delay 함수를 사용해 비동기처리를 하세요.
  await delay(t);
  mes.innerText += `${name} finished at ${startTime + t}\n`;

  // 3. 완주한 후의 시간을 반환하세요.
  return startTime + t;
}

btn.addEventListener("click", () => {
  triathlon().then((param) => {
    mes.innerText += `total time : ${param}`;
  });
});
```

3번 실습

Hint

```
//randomuser.me 라는 API는 무작위로 생성된 사용자의 프로필 이미지를 포함해 디테일한 정보까지 어디서든 요청 가능합니다.

document.getElementById("myBtn").addEventListener("click", getData);

function getData() {
  // console.log('test');

  // 1. fetch()로 API를 이용합니다. promise를 지우고 fetch를 이용해보세요
  // fetch
  // 그후 json객체로 바꿔주시고
```

```

// 그 후 처리된 json 객체의 result를 user 변수에 넣어주세요 (data.results)
Promise.resolve().then((data) => {
  let user = data.results;

  // 사용자 정보를 요청합니다.
  let output = "<h2><center>사용자 정보 받기</center></h2>";

  // 2. forEach()를 사용해서 user의 각 데이터를 output에 추가합니다.

  document.getElementById("output").innerHTML = output;
});
}

```

▼ 실습 정답 코드

```

//randomuser.me 라는 API는 무작위로 생성된 사용자의 프로필 이미지를 포함해 디테일한 정보까지 어디서든 요청 가능합니다.

document.getElementById("myBtn").addEventListener("click", getData);

function getData() {
  // console.log('test');

  //1. fetch()로 API를 이용합니다.
  fetch("https://randomuser.me/api/?results=100")
    .then((res) => res.json())
    .then((data) => {
      let user = data.results;
      // console.log(user);

      // 사용자 정보를 요청합니다.
      let output = "<h2><center>사용자 정보 받기</center></h2>";

      // 2. forEach()를 사용해서 user의 각 데이터를 output에 추가합니다.
      user.forEach(function (lists) {
        output += `
          <div class="container">
            <div class="card mt-4 bg-light">
              <ul class="list-group">
                <li class="list-group-item list-group-item-primary"><h2>Name: ${lists.name.first}</h2></li>
                <li class="list-group-item"></li>
                <li class="list-group-item">Phone Number: ${lists.cell}</li>
                <li class="list-group-item">DOB: ${lists.dob.date}</li>
                <li class="list-group-item">Age: ${lists.dob.age}</li>
                <li class="list-group-item">Email ID: ${lists.email}</li>
                <li class="list-group-item">City: ${lists.location.city}</li>
                <li class="list-group-item">Country: ${lists.location.country}</li>

                <li class="list-group-item">PostCode: ${lists.location.postcode}</li>

              </ul>
            </div>
          </div> `;
      });
      document.getElementById("output").innerHTML = output;
    });
}

```

4번 실습



`e.preventDefault()` 를 꼭 해주세요!

Hint

```
// 자유롭게 코드를 작성하여, 예시 화면이 구현되도록 해 보세요.
// async 함수 작성 (ex. getPassword())
// 새로고침 방지

// fetch 호출, await를 한번 써보세요

// 호출이 완료되면 json 형태로 바꿔주세요

// innerHTML로 가져온 데이터를 보여주세요

const buttonElem = document.querySelector('#buttonSubmit')
const passwordElem = document.querySelector('#password')

buttonElem.addEventListener("click", getPassword)
```

▼ 실습 정답 코드

```
// 자유롭게 코드를 작성하여, 예시 화면이 구현되도록 해 보세요.
async function getPassword(e) {
  // 새로고침 방지
  e.preventDefault()

  const res = await fetch('https://randomuser.me/api/?password=upper,lower,number')
  const data = await res.json()

  passwordElem.innerHTML = data.results[0].login.password
}

const buttonElem = document.querySelector('#buttonSubmit')
const passwordElem = document.querySelector('#password')

buttonElem.addEventListener("click", getPassword)
```