

2025년 상반기 K-디지털 트레이닝

자바스크립트 기초 문법과 모듈(심화1)

[KB] IT's Your Life

자바스크립트 기초 문법과 모듈

- ✓ goodbye-1.mjs 모듈에서 goodbye 함수를 bye 이름으로 임포트하여 다음과 같이 출력되도록 작성하세요.
 - `ems/app-7.mjs`

홍길동님, 안녕히 가세요.

`ems/app-7.mjs`

```
import { goodbye as bye } from './goodbye-1.mjs';
```

```
bye('홍길동');
```

```
홍길동님, 안녕히 가세요.
```

- ✓ goodbye-1.mjs 모듈에서 hi 함수를 hello 이름으로, goodbye 함수를 bye 이름으로 임포트하여 다음과 같이 출력되도록 작성하세요.
 - `ems/app-8.mjs`

```
홍길동님, 안녕하세요?  
홍길동님, 안녕히 가세요.
```

`ems/app-8.mjs`

```
import { hi as hello, goodbye as bye } from './greeting-1.mjs';
```

```
hello('홍길동');
```

```
bye('홍길동');
```

```
홍길동님, 안녕하세요?  
홍길동님, 안녕히 가세요.
```

자바스크립트 기초 문법과 모듈

- ✓ 앞에서 작성한 `greeting-1.mjs` 모듈에 내보기한 모든 함수를 `say`라는 객체 이름으로 모두 임포트하여 다음과 같이 출력되도록 작성하세요.
 - `app-9.mjs`

```
홍길동님, 안녕하세요?  
홍길동님, 안녕히 가세요.
```

`ems/app-9.mjs`

```
import * as say from './greeting-1.mjs'; // greeting.mjs에서 내보낸 함수들을 한꺼번에 say로 받기
```

```
say.hi('홍길동');  
say.goodbye('홍길동');
```

```
홍길동님, 안녕하세요?  
홍길동님, 안녕히 가세요.
```

✓ 아래 코드를 살펴보고, greeting-2 모듈을 완성하세요.

○ `ems/greeting-2.mjs`

```
const hi = (name) => {  
  console.log(`${name}님, 안녕하세요?`);  
};  
  
const goodbye = (name) => {  
  console.log(`${name}님, 안녕히 가세요.`);  
};
```

○ `ems/app-10.mjs`

```
import say from './greeting-2.mjs';  
  
say.hi('홍길동');  
say.goodbye('홍길동');
```


ems/greeting-2.mjs

```
const hi = (name) => {  
  console.log(`${name}님, 안녕하세요?`);  
};  
  
const goodbye = (name) => {  
  console.log(`${name}님, 안녕히 가세요.`);  
};  
  
export default { hi, goodbye };
```

ems/app-10.mjs

```
import say from './greeting-2.mjs';  
  
say.hi('홍길동');  
say.goodbye('홍길동');
```

2025년 상반기 K-디지털 트레이닝

자바스크립트 기초 문법과 모듈(심화2)

[KB] IT's Your Life

- ✔ Promise를 이용해서 비동기로 '피자를 주문했습니다'를 출력하도록 작성하세요.
 - `async-2.js`
 - 에러가 있는 경우 '피자를 주문하지 않습니다.'를 출력하세요.

async-2.js

```
let likePizza = true;
const pizza = new Promise((resolve, reject) => {
  1  if (likePizza)
    resolve('피자를 주문합니다.');
```

2

```
    else
      reject('피자를 주문하지 않습니다.');
```

3

```
});
pizza
  .then(result => console.log(result))
  .catch(err => console.log(err));
// 피자를 주문합니다.
```

피자를 주문합니다.

✓ 다음과 같이 출력되도록 코드를 완성하세요.

- withoutAwait.js
- 에러 처리 코드를 포함함

```
fetch('https://jsonplaceholder.typicode.com/users')
```

```
[
  {
    id: 1,
    name: 'Leanne Graham',
    username: 'Bret',
    email: 'Sincere@april.biz',
    address: {
      street: 'Kulas Light',
      suite: 'Apt. 556',
      city: 'Gwenborough',
      zipcode: '92998-3874',
      geo: [Object]
    },
    ...
  ]
```

withoutAwait.js

```
fetch('https://jsonplaceholder.typicode.com/users')  
  .then((response) => response.json())  
  .then((data) => console.log(data))  
  .catch((err) => console.log(err));
```

```
[  
  {  
    id: 1,  
    name: 'Leanne Graham',  
    username: 'Bret',  
    email: 'Sincere@april.biz',  
    address: {  
      street: 'Kulas Light',  
      suite: 'Apt. 556',  
      city: 'Gwenborough',  
      zipcode: '92998-3874',  
      geo: [Object]  
    },  
    ...  
  },  
  ...  
]
```

✓ 앞의 코드를 **async/await** 코드로 변경하세요.

- `await.js`
- 에러 처리 코드를 포함함

```
function init() {  
  
}  
  
init();
```

```
[  
  {  
    id: 1,  
    name: 'Leanne Graham',  
    username: 'Bret',  
    email: 'Sincere@april.biz',  
    address: {  
      street: 'Kulas Light',  
      suite: 'Apt. 556',  
      city: 'Gwenborough',  
      zipcode: '92998-3874',  
      geo: [Object]  
    },  
    ...  
  },  
  ...  
]
```

await.js

```
async function init() {  
  try {  
    const response = await fetch('https://jsonplaceholder.typicode.com/users');  
    const users = await response.json();  
    console.log(users);  
  } catch (err) {  
    console.error(err);  
  }  
}  
  
init();
```

```
[  
  {  
    id: 1,  
    name: 'Leanne Graham',  
    username: 'Bret',  
    email: 'Sincere@april.biz',  
    address: {  
      street: 'Kulas Light',  
      suite: 'Apt. 556',  
      city: 'Gwenborough',  
      zipcode: '92998-3874',  
      geo: [Object]  
    },  
    ...  
  },  
  ...  
]
```