■ async / await

- 비동기 작업을 동기적으로 실행되는 것처럼 보이게 만드는 방법
- promise – then – then 방식을 간단하게 표현하는 방법

```
function fetchUser() {
  console.log('작업중...');
  return new Promise((resolve, reject) => {
    resolve('작업완료');
  });
}
const user = fetchUser();
user.then(result => console.log(result))
```

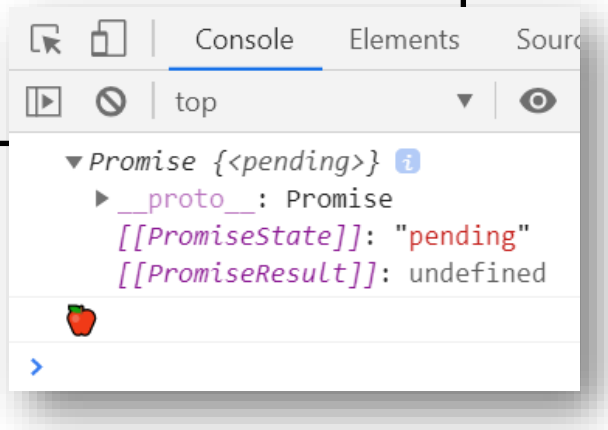
```
async function fetchUser() {
  console.log('작업중...');
  return '작업완료';
}
const user = fetchUser();
user.then(result => console.log(result))
```

■ async / await

● '사과' 를 1초 동안 가져와서 판매하는 코드 작성

- Promise / then

```
function getApple() {
  return new Promise((resolve) => {
    setTimeout(() => {
      resolve('🍎');
    }, 1000)
  });
}
function sell() {
  return new Promise((resolve) => {
    getApple().then(apple => resolve(apple));
  });
}
sell().then(res => console.log(res));
console.log(sell());  // 사용 불가
```

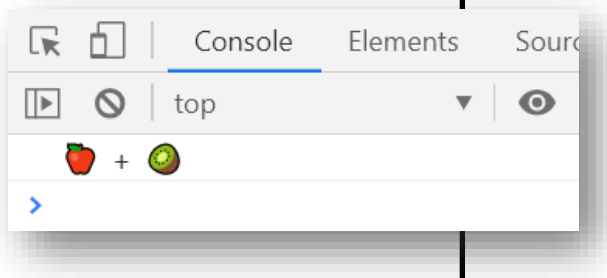


■ async / await

● **'사과' 와 '키위' 를 각각 1초 동안** 가져와서 판매하는 코드 작성

- Promise / then

```
function getApple() {
  return new Promise((resolve) => {
    setTimeout(() => {
      resolve('🍎');
    }, 1000)
  });
}
function getKiwi() {
  return new Promise((resolve) => {
    setTimeout(() => {
      resolve('🥝');
    }, 1000)
  });
}
function sell() {
  return getApple().then(apple => {
    return getKiwi().then(kiwi => `${apple} + ${kiwi}`);
  });
}
sell().then(res => console.log(res));
```

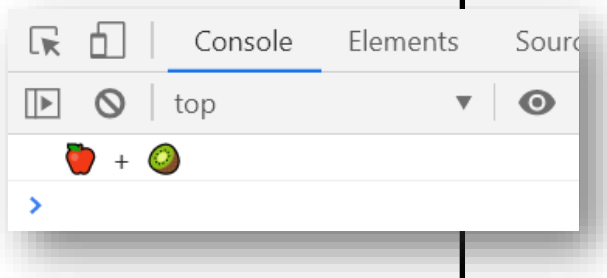


■ async / await

● '사과' 와 '키위' 를 각각 1초 동안 가져와서 판매하는 코드 작성

- async / await 적용

```
function getApple() {
  return new Promise((resolve) => {
    setTimeout(() => {
      resolve('🍎');
    }, 1000)
  });
}
function getKiwi() {
  return new Promise((resolve) => {
    setTimeout(() => {
      resolve('🥝');
    }, 1000)
  });
}
async function sell() {
  const apple = await getApple();
  const kiwi = await getKiwi();
  return `${apple} + ${kiwi}`
}
sell().then(res => console.log(res));
```

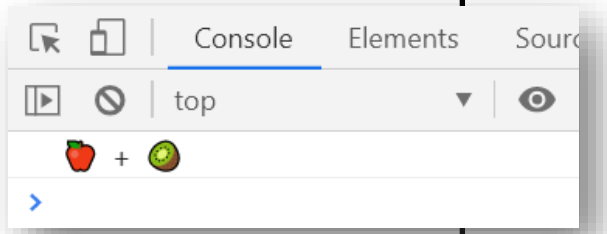


■ async / await

● **'사과' 와 '키위' 를 각각 1초 동안** 가져와서 판매하는 코드 작성

- setTimeout() 기능을 delay() 함수로 모듈화

```
function delay(ms) {
  return new Promise(resolve => setTimeout(resolve, ms));
}
async function getApple() {
  await delay(1000);
  return '🍎';
}
async function getKiwi() {
  await delay(1000);
  return '🥝';
}
async function sell() {
  const apple = await getApple();
  const kiwi = await getKiwi();
  return `${apple} + ${kiwi}`
}
sell().then(res => console.log(res));
```



■ async / await

● **'사과' 와 '키위' 를 동시에 1초 동안(병렬)** 가져와서 판매하는 코드 작성

- 함수는 즉시 실행하고 데이터를 가져오는 동안 대기

```
async function sell() {
  const applePromise = getApple();
  const kiwiPromise = getKiwi();
  const apple = await applePromise;
  const kiwi = await kiwiPromise;
  return `${apple} + ${kiwi}`;
}
```