템플릿 문자열은 백틱으로 문자열과 변수를 랩핑한 것이다.
스프레드 연산자는 배열이나 객체변수를 합치는 연산자이다.
concat()은 배열[] 을 합칠 때 사용한다. 여기서 주의할 점은 자바스크립트는 String(x) string(o) [...배열명, ...배열명]
Object.assign()은 객체{} 를 합칠 때 사용한다.
{...객체명, ...객체명}
const ['a','b','c'] = arr
const {a, b, c} = obj
추출할 때는 구조분해할당을 한다.
props 는 부모에서 자식으로 전달되는 데이터이고, immutable 이다.

hook 는 함수 컴포넌트에서 상태관리, 라이프사이클을 연동하게 하는 함수다.
effects 는 컴포넌트 안에서 데이터를 가져오거나 구독하고, DOM을 직접 조작하는 작업이다.
최초의 비동기는 ajax() 이다.

```
$.ajax({
    $.ajax({...})
})
```

콜백지옥...
new Promise(function(){}).then() 는 싱글밸류를 가진 비동기 처리 객체이며. 3가지 상태(대기, 성공, 실패)를 갖는다.
then() 함수는 성공상태 일때 작동하는 리스너이다.
catch()함수는 실패상태 이때 작동하는 리스너이다.
aync, await 의 구조

```
// async & await 적용 후
async function logName() {
  var user = await fetchUser(URL);
  if (user.id === 1) {
    console.log(user.name);
  }
}
const makeRequest = () =>
  getJSON()
    .then(data => {
      console.log(data)
      return "done"
    })
makeRequest()
const makeRequest = async () => {
  console.log(await getJSON())
  return "done"
}
makeRequest()
const makeRequest = async () => {
  try {
    // this parse may fail
    const data = JSON.parse(await getJSON())
    console.log(data)
  } catch (err) {
    console.log(err)
  }
}
```

최초의 비동기 $ajax 이다. (함수이다)

↓ 콜백지옥

Promise는 성공 보류를 가진 비동기 처리 객체이며, 3가지 상태를 가진다. (대기/성공/실패)

then() 함수는 성공상태 일때 작동 리스너

catch()는 실패 〃

new Promise()

↓ ES6 (2015年)

문제점: 상태를 가진다 → 메모리의 리소스를 잡식

+인공지능의 탄생.

async / await

함수이다.

바닐라스크립트

```
async function logName() {
  var user = await fetchUser('domain.com/users/1');
  if (user.id === 1) {
    console.log(user.name);
  }
}
```

→ URL

동기코드에 async/await 만 붙이면됨.

기존 axios

```
const fetchUsers = () => {

    axios.get(`http://localhost:8080/users/list`)
    .then(res => {
        console.log(res)
        setUsers(...users ,res.data)

    })
    .catch(err => alert(err));
};
useEffect(() =>{
    fetchUsers();
},[]);
```

프로미스를 사용한다 (상태o)

↓

async/await 사용하라!