

Chapter 4. 엘리먼트 렌더링

교재: 처음만난 리액트 (저자: 이인제, 한빛출판사)



Contents

- CHAPTER 4: 엘리먼트 렌더링

- 4.1 엘리먼트에 대해 알아보기

- 4.2 엘리먼트 렌더링하기

- 4.3 렌더링된 엘리먼트 업데이트하기

- 4.4 (실습) 시계 만들기

SECTION 4.1 엘리먼트에 대해 알아보기

◦ 엘리먼트의 정의

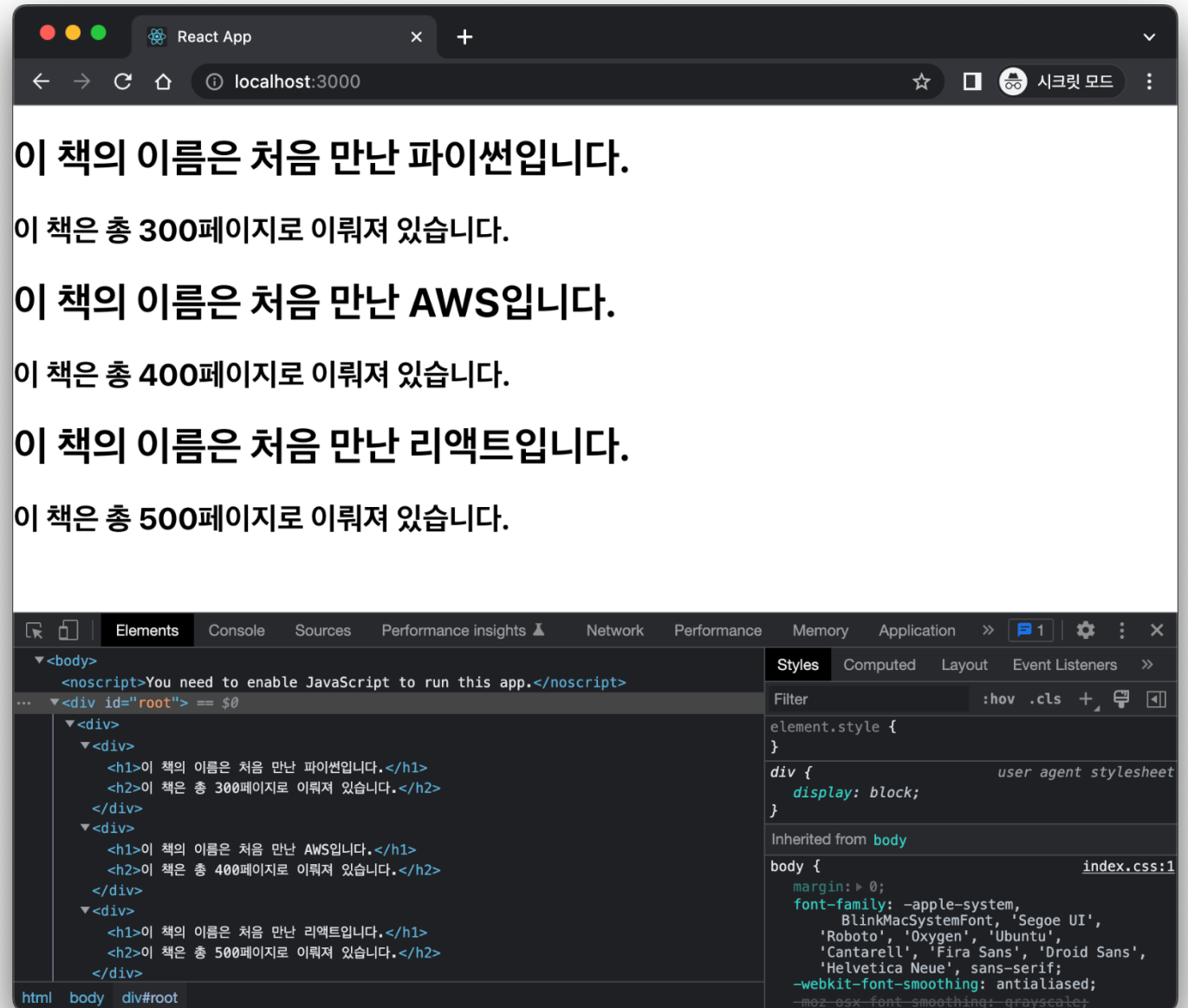
- 엘리먼트 - 어떤 물체를 구성하는 성분

- **Elements are the smallest building blocks** of React apps.

- 리액트 앱을 구성하는 가장 작은 블록들

SECTION 4.1 엘리먼트에 대해 알아보기

- DOM 엘리먼트
 - HTML 요소



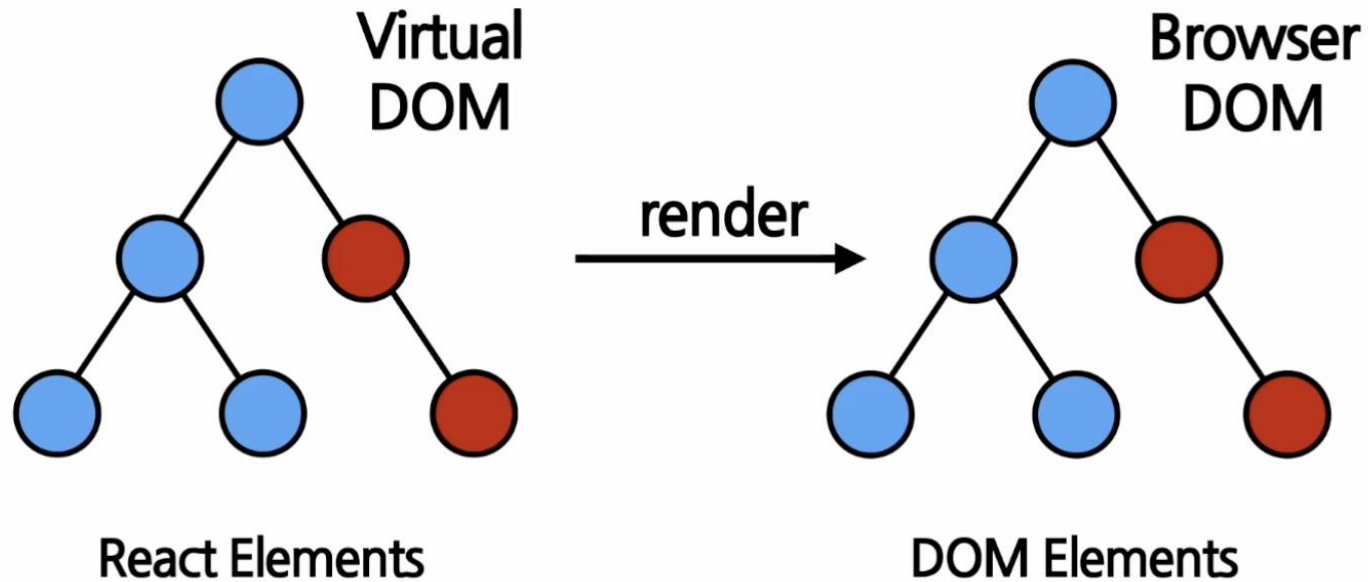
SECTION 4.1 엘리먼트에 대해 알아보기

◦ 리액트 엘리먼트

- 화면에 나타나는 내용을 기술한 자바스크립트 객체 -> **Descriptor**
- Descriptor가 최종적으로 나타나는 형태는 DOM 엘리먼트
-> DOM과의 통일성을 위해 **엘리먼트**라는 명칭으로 변경

SECTION 4.1 엘리먼트에 대해 알아보기

- 리액트 엘리먼트, DOM 엘리먼트 관계



- 리액트 엘리먼트는 DOM 엘리먼트의 가상 표현
- 엘리먼트는 화면에 보이는 것들을 기술
 - > 이를 토대로 실제 화면에서 보게 되는 DOM 엘리먼트가 생성됨

SECTION 4.1 엘리먼트에 대해 알아보기

- 엘리먼트 - JSX 코드

```
1 const element = <h1>Hello, world!</h1>;
```

- 리액트의 createElement() 함수에 의해 리액트 엘리먼트가 생성됨

SECTION 4.1 엘리먼트에 대해 알아보기

- 엘리먼트의 생김새
 - 리액트 엘리먼트는 자바스크립트 객체 형태로 존재
 - 컴포넌트 유형, 속성, 자식(Children)에 대한 정보를 포함하는 자바스크립트 객체

```
1 {  
2   type: 'button',  
3   props: {  
4     className: 'bg-green',  
5     children: {  
6       type: 'b',  
7       props: {  
8         children: 'Hello, element!'  
9       }  
10    }  
11  }  
12 }
```

리액트 엘리먼트

```
1 <button class='bg-green'>  
2   <b>  
3     Hello, element!  
4   </b>  
5 </button>
```

DOM 엘리먼트

SECTION 4.1 엘리먼트에 대해 알아보기

- 엘리먼트의 생김새 - 리액트 컴포넌트
 - 리액트 엘리먼트는 자바스크립트 객체 형태로 존재

```
1 {  
2   type: Button,  
3   props: {  
4     color: 'green',  
5     children: 'Hello, element!'  
6   }  
7 }
```

```
1 React.createElement(  
2   type,  
3   [props],  
4   [...children]  
5 )
```

SECTION 4.1 엘리먼트에 대해 알아보기

- createElement() 함수의 동작 과정

```
1 function Button(props) {
2   return (
3     <button className={`bg-${props.color}`}>
4       <b>
5         {props.children}
6       </b>
7     </button>
8   )
9 }
10
11 function ConfirmDialog(props) {
12   return (
13     <div>
14       <p>내용을 확인하셨으면 확인 버튼을 눌러주세요.</p>
15       <Button color='green'>확인</Button>
16     </div>
17   )
18 }
```

SECTION 4.1 엘리먼트에 대해 알아보기

- createElement() 함수의 동작 과정 - ConfirmDialog

```
1 {
2   type: 'div',
3   props: {
4     children: [
5       {
6         type: 'p',
7         props: {
8           children: '내용을 확인하셨으면 확인 버튼을 눌러주세요.'
9         }
10      },
11      {
12        type: Button,
13        props: {
14          color: 'green',
15          children: '확인'
16        }
17      }
18    ]
19  }
20 }
```

SECTION 4.1 엘리먼트에 대해 알아보기

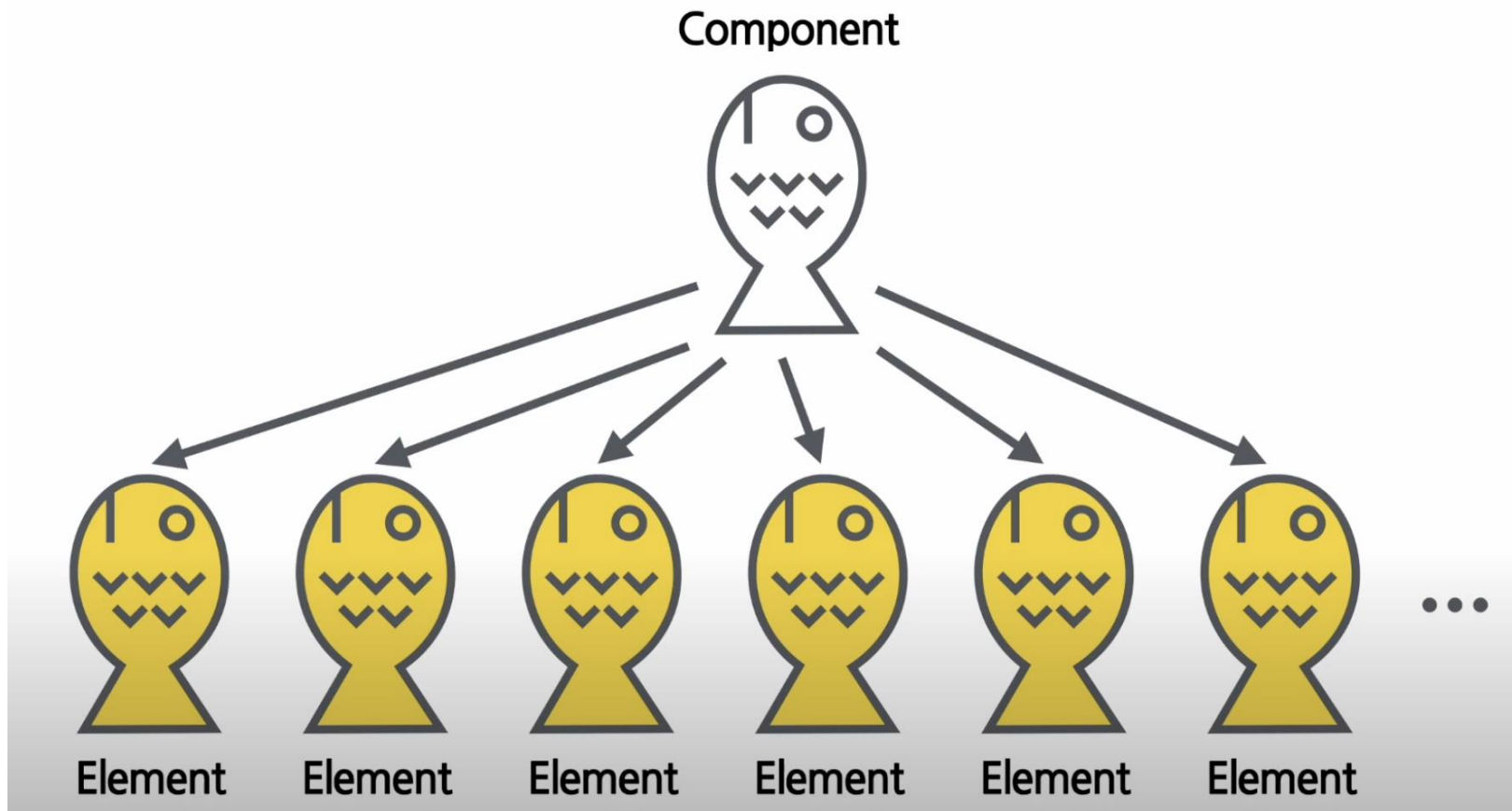
- createElement() 함수의 동작 과정

```
1 {
2   type: 'div',
3   props: {
4     children: [
5       {
6         type: 'p',
7         props: {
8           children: '내용을 확인하셨으면 확인 버튼을 눌러주세요.'
9         }
10      },
11      {
12        type: 'button',
13        props: {
14          className: 'bg-green',
15          children: {
16            type: 'b',
17            props: {
18              children: '확인'
19            }
20          }
21        }
22      }
23    ]
24  }
25 }
```

SECTION 4.1 엘리먼트에 대해 알아보기

- 엘리먼트의 특징

- 불변성(immutable) - 변하지 않는 성질
 - 엘리먼트 생성 후에는 children이나 attributes를 바꿀 수 없음

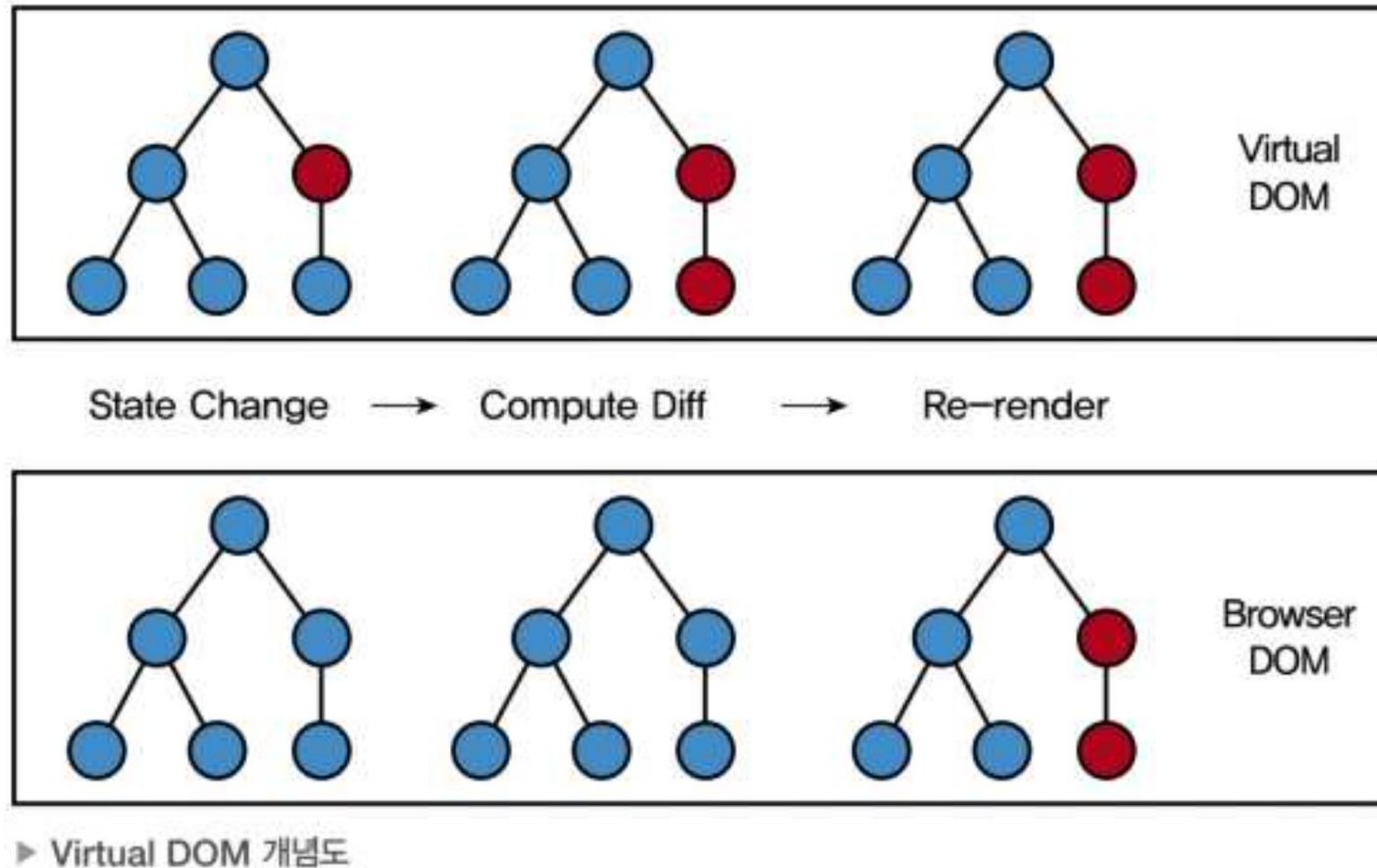


SECTION 4.1 엘리먼트에 대해 알아보기

◦ 엘리먼트의 특징 - 불변성

- 화면에 변경된 엘리먼트를 보여주려면?

-> 새로운 엘리먼트를 만들어서 기존 엘리먼트를 바꿔치기



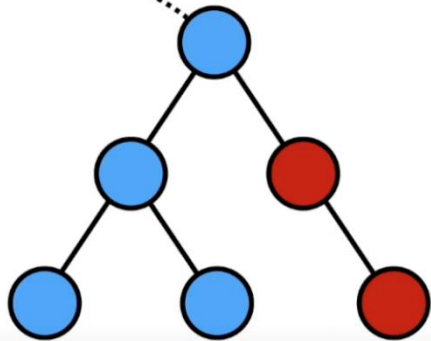
SECTION 4.2 엘리먼트 렌더링하기

◦ Root DOM Node

```
<div id="root"></div>
```


- 이 div 태그 안에 리액트 엘리먼트들이 렌더링 되며, 리액트 DOM에 의해 관리됨
- 리액트만으로 만들어진 앱은 단 하나의 Root DOM Node를 가짐

Root DOM Node



SECTION 4.2 엘리먼트 렌더링하기

- Root DOM Node에 엘리먼트 렌더링하기



```
const element = <h1>안녕, 리액트!</h1>;  
const root = ReactDOM.createRoot(document.getElementById('root'));  
root.render(element);
```

- 엘리먼트가 렌더링되는 과정은 Virtual DOM에서 실제 DOM으로 이동하는 과정

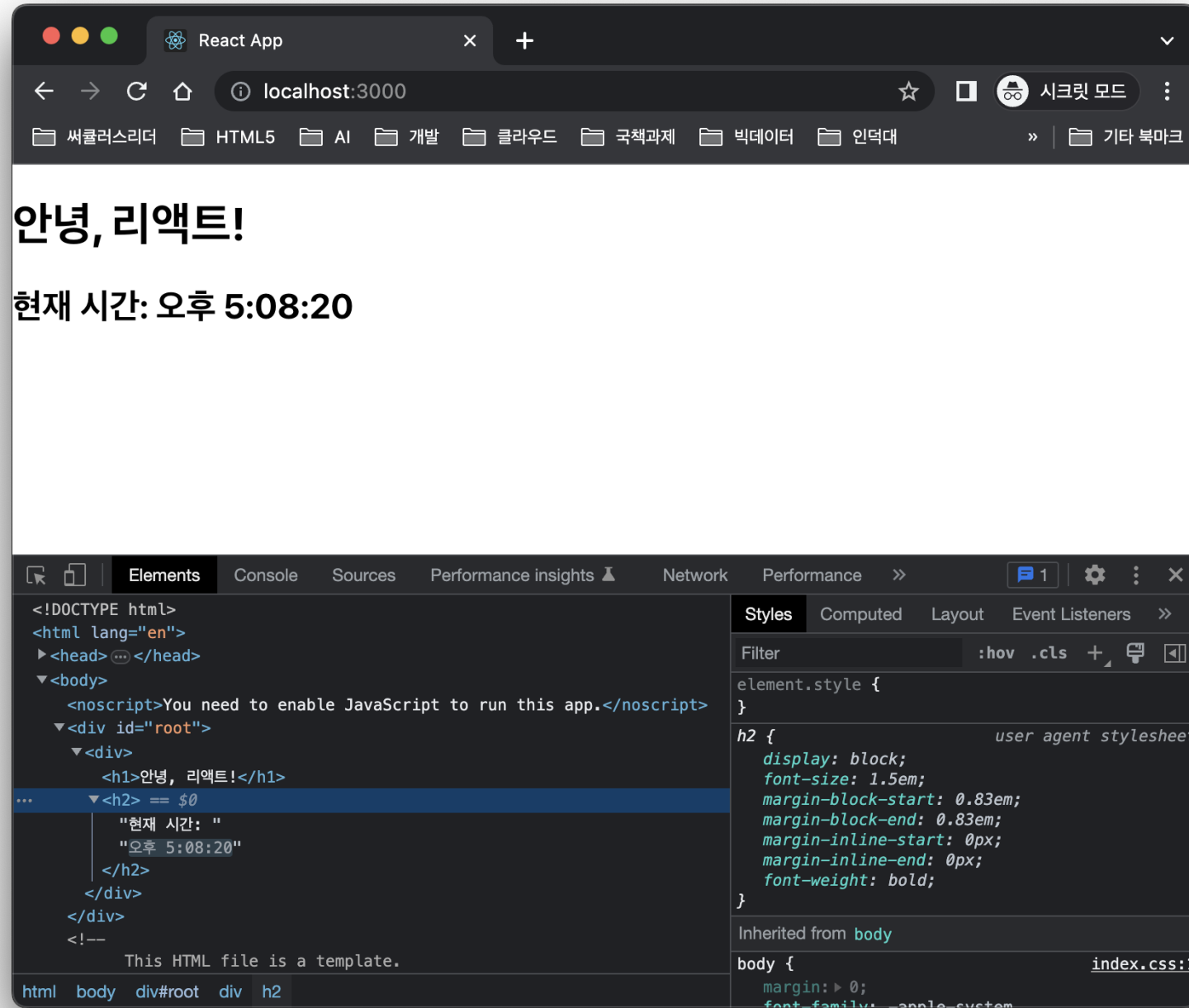
SECTION 4.3 렌더링된 엘리먼트 업데이트하기

- 엘리먼트의 불변성
 - 새로운 엘리먼트를 생성해서 기존 엘리먼트를 바꿔치기

```
function tick() {  
  const element = (  
    <div>  
      <h1>안녕, 리액트!</h1>  
      <h2>현재 시간: {new Date().toLocaleTimeString()}</h2>  
    </div>  
  );  
  
  const root = ReactDOM.createRoot(document.getElementById('root'));  
  root.render(element);  
}  
  
setInterval(tick, 1000);
```

SECTION 4.3 렌더링된 엘리먼트 업데이트하기

◦ 실행 결과



PRACTICE 4.4 시계 만들기

- VS Code - 04 폴더 생성
- Clock.jsx

```
import React from "react";

function Clock(props) {
  return (
    <div>
      <h1>안녕, 리액트!</h1>
      <h2>현재 시간: {new Date().toLocaleTimeString()}</h2>
    </div>
  );
}

export default Clock;
```

PRACTICE 4.4 시계 만들기

- index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';

import Clock from './04/Clock';

const root = ReactDOM.createRoot(document.getElementById('root'));
setInterval(() => {
  root.render(
    <React.StrictMode>
      <Clock />
    </React.StrictMode>
  );
}, 1000);

reportWebVitals();
```

[요약]

◦ 엘리먼트

▪ 엘리먼트의 정의

- 리액트 앱의 가장 작은 빌딩 블록들
- 화면에 나타나는 내용을 기술하는 자바스크립트 객체
- 리액트 엘리먼트는 DOM 엘리먼트의 가상 표현

▪ 엘리먼트의 생김새

- 엘리먼트는 자바스크립트 객체 형태로 존재
- 컴포넌트 유형, 속성, 모든 자식에 대한 정보를 포함하고 있는 자바스크립트 객체

▪ 엘리먼트의 특징

- 불변성
- 엘리먼트 생성 후에는 자식이나 속성을 바꿀 수 없음

[요약]

- 엘리먼트 렌더링하기
 - 렌더링하기 위해 ReactDOM.createRoot(); render() 함수 사용
 - 리액트 엘리먼트를 HTML 엘리먼트에 렌더링하는 역할
 - 렌더링되는 과정은 Virtual DOM에서 실제 DOM으로 이동하는 과정
- 렌더링된 엘리먼트 업데이트하기
 - 엘리먼트는 한번 생성되면 바꿀 수 없기 때문에 다시 생성해야 함
 - 기존 엘리먼트를 변경하는 것이 아니라 새로운 엘리먼트를 생성해서 바꿔치기