React Basic

개요

React는 사용자 인터페이스를 구성하기 위한 라이브러리이다. React는 다른 라이브러리와 함께 특정 환경의 사용자 인터페이스를 구성하는 데 사용된다. (React - Web, React Native - Mobile, React360 - VR) (~~근데, 메타 홈페이지에서 보면 Framework라고 명명되어 있기는 함.~~ 만약, React Framework라고 지칭한다면 통상적으로 React DOM을 의미한다.)

React는 JSX(Javascript와 XML)라는 HTML-in-Javascript 문법을 사용한다. 개발자가 직접 DOM에 접근하여 UI를 그리는 것은 비생산적이며 방대한 코드를 작성해야 하는 불편함이 있는데 React는 가상의 DOM과 React DOM을 활용하여 개발자가 UI를 그리는 데 집중할 수 있게 도와주며, 코드를 더욱 직관적이게 작성할 수 있도록 도와준다.

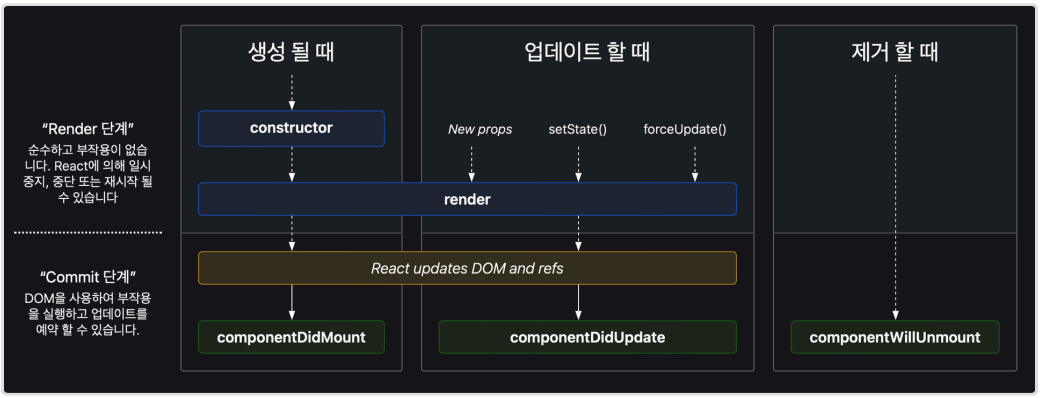
Javascript를 활용해서 DOM 직접 제어로 를 그리기 위한 함수 예시

```
function buildTodoItemEl(id, name) {
  const item = document.createElement("li");
  const span = document.createElement("span");
  const textContent = document.createTextNode(name);

  span.appendChild(textContent);

  item.id = id;
  item.appendChild(span);
  item.appendChild(buildDeleteButtonEl(id));

  return item;
}
```

React를 활용한 컴포넌트 생성 함수 예시

```
function AuthorCredit(props) {
  return (
    <figure>
      <img src={props.src} alt={props.alt} />
      <figcaption>{props.byline}</figcaption>
    </figure>
  );
}
```

React의 수명주기



React Lifecycle Methods diagram

Components

웹에서는 HTML을 통해 `<h1>`, `<li>` 같은 태그들을 사용하여 문서 구조를 만든다.

```
<article>
  <h1>My First Component</h1>
  <ol>
    <li>Components: UI Building Blocks</li>
    <li>Defining a Component</li>
    <li>Using a Component</li>
  </ol>
</article>
```

React를 사용하면 HTML, CSS, JavaScript를 사용하여 재사용 가능한 UI Component를 만들 수 있다.

```
<PageLayout>
  <NavigationHeader>
    <SearchBar />
    <Link to="/docs">Docs</Link>
  </NavigationHeader>
  <Sidebar />
  <PageContent>
    <TableOfContents />
    <DocumentationText />
  </PageContent>
</PageLayout>
```

컴포넌트 정의

React Component는 JSX 문법을 사용한다. JSX는 마크업으로 뿌릴 수 있는 JavaScript이다. (문법이 HTML과 비슷하지만 세세하게 다른 부분이 몇 가지 있다.)

```
export default function Profile() {
  return (
    <img
      src="https://i.imgur.com/MK3eW3Am.jpg"
      alt="Katherine Johnson"
    />
  )
```

```
}
```

- React component는 일반 JavaScript 함수이지만 이름은 대문자로 시작해야 한다. 그렇지 않으면 동작하지 않는다.
- HTML 태그와 똑같이 생겼지만 실제로는 JSX로 JavaScript 코드이다. JavaScript 코드 안에 마크업을 삽입할 수 있다.
- JSX는 반드시 하나의 태그 안에 묶여있어야 사용이 가능하다. 여러 태그를 리턴할 수 없다.

컴포넌트 사용

위에서 정의한 Profile이라는 React Component를 HTML 태그처럼 사용할 수 있다.

```
export default function Gallery() {
  return (
    <section>
      <h1>Amazing scientists</h1>
      <Profile />
      <Profile />
      <Profile />
    </section>
  );
}
```

브라우저에 표시되는 내용

대소문자 차이가 있다.

- `<section>`은 소문자로 시작하기 때문에 React가 HTML 태그라고 이해한다.
- `<Profile>`은 대문자로 React가 React component라고 이해한다.

```
<section>
  <h1>Amazing scientists</h1>
  <img src="https://i.imgur.com/MK3eW3As.jpg" alt="Katherine Johnson" />
  <img src="https://i.imgur.com/MK3eW3As.jpg" alt="Katherine Johnson" />
  <img src="https://i.imgur.com/MK3eW3As.jpg" alt="Katherine Johnson" />
</section>
```

컴포넌트 정의 주의 사항

Component 정의 함수 안에서 다른 Component를 렌더링할 수 있지만, 정의를 중첩하는 것은 피해야 한다.

```
export default function Gallery() {
  // 🔴 절대 컴포넌트 안에 다른 컴포넌트를 정의하면 안 됩니다!
  function Profile() {
    // ...
  }
  // ...
}
```

자식 컴포넌트에서 부모 컴포넌트의 일부 데이터가 필요한 경우라면, 정의를 중첩하지 않고 props로 전달하는 것이 필요하다.

```
export default function Gallery() {
  // ...
}

// ✅ 최상위 레벨에서 컴포넌트를 선언합니다
```

```
function Profile() {
  // ...
}
```

Import, Export

정의한 Component를 다른 파일로 옮길 수 있다.

- JS 파일을 생성한다.
- 새로 만든 파일에서 Component를 export 한다. (default export, named export 방식이 있다.)
- Component를 사용할 파일에서 import 한다. (적절한 방식을 선택해서 default, named import를 한다.)

```
import Gallery from './Gallery.js';

export default function App() {
  return (
    <Gallery />
  );
}
```

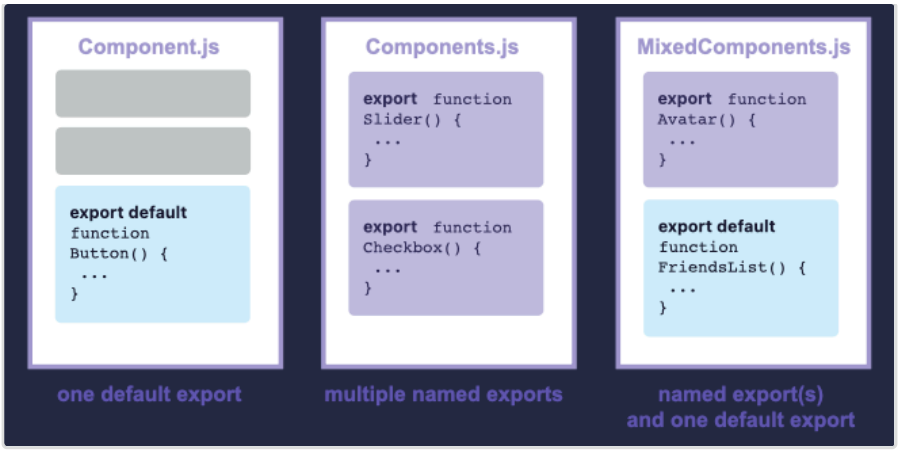
```
function Profile() {
  ...
}

export default function Gallery() {
  return (
    <section>
      <h1>Amazing scientists</h1>
      <Profile />
      <Profile />
      <Profile />
    </section>
  );
}
```

- App.js :
  - Default 방식으로 `Gallery`를 Gallery.js 파일로부터 import 한다.
  - Root App component를 default 방식으로 export 한다.
- Gallery.js :
  - Default 방식으로 `Gallery` Component를 export 한다.
  - Profile Component를 정의하고 해당 파일에서만 사용되기 때문에 export 되지 않는다.

> `import Gallery from './Gallery';`
>
> React 에서는 `'./Gallery.js'` 또는 `'./Gallery'` 둘 다 사용할 수 있다. 전자는 native ES module 사용 방법에 더 가깝다.

Default와 Named Export

보통 JavaScript에서 default와 named export 라는 두 가지 방법으로 함수를 export 한다. 한 파일에 하나의 default export만 존재할 수 있고 named export는 여러 개 존재할 수 있다.



Export 하는 방식에 따라 import 하는 방식이 정해져 있다.

Syntax	Export 구문	Import 구문
Default	`export default function Button() {}`	`import Button from './button.js';`
Named	`export function Button() {}`	`import { Button } from './button.js';`

Default import를 사용하는 경우 import 단어 뒤에 다른 이름으로 값을 가져올 수도 있다. 예를 들어, `import Something from './button.js'` 라고 선언해도 같은 default export 함수를 가져온다. 반대로 named import를 사용할 때는 양쪽 파일에서 사용하고자 하는 함수의 이름이 같아야 하기 때문에 named import로 부른다.

참고 자료 :

첫 번째 컴포넌트 – React

컴포넌트 import 및 export 하기 – React