프론트엔드 면접문제\_강현주\_20200702

1. 컴포넌트 설계가 갖는 장점은 무엇인가. 컴포넌트란 무엇인가. 컴포넌트 설계는 어떻게 해야 하는가. 컴포넌트의 리렌더링은 언제 이루어지는가.

* **컴포넌트의 정의** : 리액트에서 가장 중요한 구성요소이자 자바스크립트 함수이다. 리액트로 만들어진 앱, 웹을 이루는 가장 최소한의 단위가 컴포넌트이며 리액트로 만든다는 것은 곧 작고 단단한 컴포넌트들을 만들고 이 컴포넌트들을 유기적으로 연결한다는 것을 뜻한다.

컴포넌트라는 단위가 갖는 의미는 곧 데이터를 입력받아 DOM Node를 출력하는 함수이다. 이 때 입력받는 데이터는 Props나 State이다. 즉, 임의의 Props라 불리는 인풋을 받아서 화면에 무엇이 보여질 지 기술(describing)하는 리액트 엘리먼트를 반환한다.

리액트의 컴포넌트에는 클래스 컴포넌트, 함수형 컴포넌트 두가지 종류의 컴포넌트가 있다.

* **컴포넌트의 설계와 장점** : 컴포넌트는 위에서 말했듯 웹 사이트를 구성하고 있는 요소이며 HTML로 비유하면 <header>, <footer> 등 웹 사이트 페이지별로 들어간 공통된 영역이다. 이들을 미리 리액트로 컴포넌트화하면 페이지별로 필요한 컴포넌트를 조립하는 방식으로 쉽고 빠르게 결과물을 만들 수 있다.

웹사이트의 특정 요소를 따로 분리하여 Header 컴포넌트, Main 컴포넌트 등으로 작업하고, 이를 조립해 나가는 방식으로 만든다면 보다 효율적으로 코드를 관리할 수 있고 DRY (Don’t Repeat Yourself)를 유지, 재사용 가능한 컴포넌트 만들기, 또 컴포넌트가 무엇을 하는지 명확하게 이해할 수 있다.

* **컴포넌트의 리렌더링이 이루어지는 시점**

1. 자신의 상태 (state)가 변경될 때
2. 부모 컴포넌트가 리렌더링 될 때
3. 자신이 전달받은 props가 변경될 때
4. forceUpdate 함수가 실행될 때

출처 :

1. <https://brunch.co.kr/@brunch92ny/34>
2. <https://code-masterjung.tistory.com/m/67>

2. 리액트와 뷰의 차이점과 공통점은 무엇인가.

* **공통점**

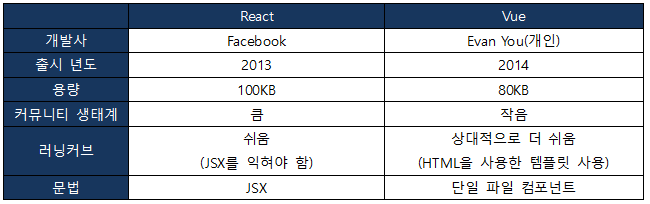
1. **컴포넌트 기반** : 컴포넌트 단위로 변화사항을 감지하고 렌더링이 필요한 부분만 바꾸어주는 기능은 양쪽의 프레임워크가 목적론적으로나 방법론적으로나 동일하다. 둘다 virtual DOM을 기반으로 하며, tree알고리즘을 통해 변화사항을 찾아낸다.
2. **PROPS, STATE, LIFECYCLE** : 둘 다 컴포넌트 단위로 코드를 작성하는 친구들이다보니, 컴포넌트의 기본적인 구성요소도 겹치는 것들이 있다. 대표적인게 props와 state, 그리고 lifecycle 이다. 세부적인 명칭이나 사용법 등은 물론 조금씩 다르긴 하지만, 지향점은 같다.
3. **라우터, 전역 상태관리도구** : SPA를 통해 여러 페이지를 보여주기 위한 라우터 역시 리액트와 뷰 둘다 존재한다.
4. **SSR Framework** : 둘다 서버사이드 렌더링용 웹서버 프레임워크가 있다. webpack 기반 자바스크립트이다보니 둘다 nodeJS 기반이다.

* **차이점**
* **리액트 :**

1. **거대한 커뮤니티** : 페이스북의 지원을 받는 만큼 지속적인 버전 관리가 이루어지고, React 사용자가 많아 다양한 레퍼런스, 확장 라이브러리가 많다. 또한 이미 React를 사용하고 있는 개발 프로젝트가 많기 때문에 개발자 채용 시, React를 다룰 줄 안다는 것이 장점으로 작용할 수 있다.
2. **JSX 기반 컴포넌트 구문** : 템플릿 방식으로 관리하는 Vue와는 달리, React는 JSX 코드로 컴포넌트를 작성하고 컴포넌트의 상태(State)를 변화시키지 않고 관리한다. 변화가 일어나면 실제 브라우저의 DOM에 새로운 것을 적용하는 것이 아니라, 자바스크립트로 이루어진 Virtual DOM에 렌더링을 하고 기존의 DOM과 비교하여 변화가 일어난 곳만 업데이트 한다.템플릿은 재구조화가 어렵고, 에러에 취약하지만, React는 상태를 고유 속성대로 유지하기 때문에 대규모 프로젝트에서도 관리가 수월하고 테스트에도 적합하다.

* **뷰 :**

1. **단일 파일 컴포넌트(Single File Component)** : Vue는 단일 파일 컴포넌트 기반의 프레임워크이다. 웹의 뷰(view)를 구성하는 요소인 HTML, CSS, JavaScript 코드를 .vue 확장자를 가진 하나의 파일에 모두 정의하는 방식. 이러한 관리 방식은 적당한 규모의 프로젝트에서 관리의 생산성을 높이고, 협업을 수월하게 한다는 장점이 있다.
2. **HTML 기반 템플릿 구문** : Vue는 개발한 프론트엔드 파일을 사용자가 볼 수 있도록 브라우저 화면에 렌더링하는 과정에 템플릿이란 문법을 사용한다. 이 템플릿을 구성하는 문법이 HTML 기반으로 이뤄져 있어 배우기 쉽다는 특징이 있다.



출처 :

1. <https://velopert.com/3236>
2. <https://velog.io/@vraimentres/react-vs-vue-1>
3. <http://library.gabia.com/contents/infrahosting/8284>

3. STATE와 PROPS를 설명하시오.

리액트 컴포넌트에서 다루는 데이터는 두개로 나뉜다. 그것이 바로 PROPS와 STATE이다.

* PROPS : 부모 컴포넌트가 자식 컴포넌트에게 주는 값이다. 자식 컴포넌트에서는 PROPS를 받아오기만하고, 받아온 PROPS를 직접 수정할 수는 없다.
* STATE : PROPS와 달리 내부에서 선언하며 내부에서 값을 읽고 변경할 수 있다. 기본값을 설정해둬야 한다.

* PROPS와 STATE로 나눈 이유 : 개발자들에게 명확한 관념 모델을 제공한다. 관념 모델은 무엇이 어떻게 동작하는 지 이해할 때 진행되는 일련의 사고 프로세스를 의미하며 즉, 논리적으로 이치에 맞는 사고모델을 제공한다는 것이다.

PROPS의 변경, STATE가 없을 때의 값 등등 이런 질문들에 고민할 필요 없이 컴포넌트 간에는 무조건 PROPS를 통해서만 데이터를 주고받고 PROPS는 컴포넌트 내부에서 변경되지 않으므로 위/아래, 양쪽에 대해 동시에 고민할 필요가 없고 아래 한쪽 방향과 자기자신에 대해서만 고민하면 되는 것이다.

지금 컴포넌트에서 필요한 값이 PROPS인지 STATE인지 판단하고 어느 것과 관련이 있는지 이 값을 어떤 컴포넌트에 어떻게 넘겨줄지 생각하여 코드를 작성하면 컴포넌트를 완성할 수 있다.

출처 :

1. <https://kim6394.tistory.com/117>
2. <https://medium.com/little-big-programming/react%EC%9D%98-%EA%B8%B0%EB%B3%B8-%EC%BB%B4%ED%8F%AC%EB%84%8C%ED%8A%B8%EB%A5%BC-%EC%95%8C%EC%95%84%EB%B3%B4%EC%9E%90-92c923011818>