

React Study

목 차

1. 스터디소개
2. 향후진행방향
3. What is React – concept
4. React Feature
5. Hello React

2018. 10. 17

1. 스터디 소개

- 리액트 공부하는 초급자 스터디
- 리액트를 전혀 몰랐던 개발자들이 주로 관련책과 인터넷으로 공부한내역 공유
- 리액트 이외에도 스터디인원들의 기술스펙 및 관심기술 공유
- 진행자가 참고했던 책
- 리액트교과서 공부후 넷플릭스 영화소개 예제 구현
- 리액트를 다루는 기술 공부후 일정관리투두 CRUD 예제 구현

- 리액트강좌 목차참고

강좌내용	1 강 - 3 시간 React 개요 - React 소개 - JSX 및 es6, es7 소개 - 개발환경 구축
	2 강 - 3 시간 React 컴포넌트 - React 컴포넌트 소개 - Prop 과 관련 함수 - State 와 관련 함수
	3 강 - 3 시간 React 이벤트 및 Form - 이벤트 바인딩 - Form 다루기 - mix-in
	4 강 - 3 시간 React Flux - React lifecycle - Flux - 공개된 컴포넌트 사용
	5 강 - 3 시간 React Routing - Routing

fast campus		
회차	주제	세부내용
1회차	리액트 시작하기	리액트를 왜 쓸까? 소개 및 환경설정 Component의 이해 JSX, props, state, 이벤트 관리 Lifecycle API
2회차	프로젝트 만들기	리액트 컴포넌트 디버깅 Todo List 구현하기 컴포넌트 스타일링 CSS Module, Sass, Styled-components
3회차	리액트 활용하기	컴포넌트 스타일링 / API 연동 실습 NASA 오늘의 사진 갤러리 프로젝트 구현하기 컴포넌트 최적화 방법 알아보기 불변성 유지 라이브러리 (i) Immutable.js 불변성 유지 라이브러리 (ii) Immer.js
4회차	Redux 리덕스	Prettier + ESLint 사용하기 Higher Order Component(HoC) 알아보기 리덕스 기초 학습하기

2. 향후스터디 진행방향

- 1회차 : React Intro, ES6기초study const
- 2회차 : JSX, CRA, webpack, ES6 배열, 오브젝트, map(), VSC 디버그
- 3회차 : React state, props, ES6 화살표함수, SCSS스타일링
- 4회차 : React LifeCycle, ES6 스프레드 레스트, 성능이슈확인(크롬 performance)
- 5회차 ~ 8회차 : 영화조회사이트 구현 - 리액트교과서 참고, Node
- 9회차~ 12회차 : To Do project CRUD - 리액트를다루는기술 참고, Node
- 결론적으로 react 초급탈출하는게 목표이며
- 탈출과정에서 개인포트폴리오 2개를 만드는것입니다.
- 내용은 진행자가 임의로 변경할 수 있습니다.

3. What is React - concept

- 리엑트는 자바스크립트 라이브러리이다. (깃헙에서 별 십만개 이상 받은 초인기) 템플릿언어는 아니다.
기존에 보여줄 html같은것에 변경부분처리하는 것을 템플릿
- 리엑트는 프레임워크는 아니다. (앵귤라는 프레임웍이다)
- 리엑트는 페이스북에서 만들고 사용중인 오픈소스 프로젝트이다.
- 리엑트는 프론트엔드 UI를 만드는데 사용한다.
- 리엑트는 MVC application의 View를 만드는데 사용한다.
- UI를 컴포넌트 형식으로 빠르고 재사용가능하고 능률적으로 만들수 있게 해준다.
- 리엑트는 순수한 자바스크립트를 기반으로 컴포넌트를 구현하여 재사용한다.

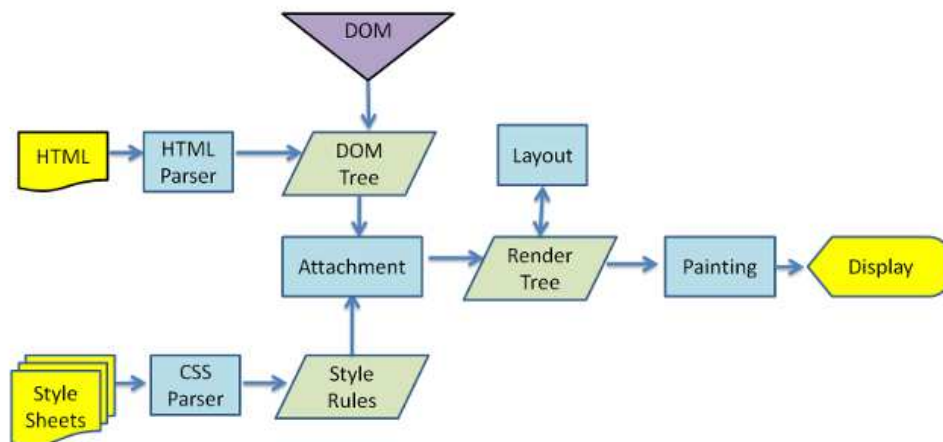
4. React Feature – virtual DOM

- Virtual DOM을 활용함
- 빠르다? 충분히 빠르고 최적화를 생산적으로 함
- 아래 참조는 <https://velopert.com/3236>

브라우저의 Workflow

이해를 잘 하기 위해선 브라우저가 어떻게 작동하는지 알아볼 필요가 있어요.

NOTE: 다음 다이어그램과 앞으로 나올 설명은 Webkit 엔진에서 사용하는 용어들을 사용했어요. 작동방식은 대부분의 브라우저에서 비슷합니다. 미묘한 차이가 있을 뿐이에요.



- 디지털 시계를 간단히 구현해서 왜 빠른것인지 코드로 한번 보겠습니다.

Virtual DOM 을 만나보세요

자 이제 DOM 을 조작했을 때 어떤 작업이 이뤄지는지 알겠죠? DOM에 변화 생기면, 렌더 트리를 재생성하고 (그러면 모든 요소들의 스타일이 다시 계산됩니다) 레이아웃을 만들고 페인팅을 하는 과정이 다시 반복되는거죠.

복잡한 SPA(싱글 페이지 어플리케이션)에서는 DOM 조작이 많이 발생해요. 그 뜻은 그 변화를 적용하기 위해 브라우저가 많이 연산을 해야한단 소리고, 전체적인 프로세스를 비효율적으로 만듭니다.

자, 이 이부분에서 Virtual DOM 이 빛을 발합니다! 만약에 뷰에 변화가 있다면, 그 변화는 실제 DOM에 적용되기전에 가상의 DOM에 먼저 적용시키고 그 최종적인 결과를 실제 DOM으로 전달해줍니다. 이로써, 브라우저 내에서 발생하는 연산의 양을 줄이면서 성능이 개선되는 것이지요.

5. React 라이브러리 확인

- 리액트가 사용하는 라이브러리를 확인
- React
- React DOM

```
/**
 * Create and return a new ReactElement of the given type.
 * See https://reactjs.org/docs/react-api.html#createelement
 */
```

```
function createElement(type, config, children) {
  var propName = void 0;

  // Reserved names are extracted
  var props = {};

  var key = null;
  var ref = null;
  var self = null;
  var source = null;

  if (config != null) {
    if (hasValidRef(config)) {
      ref = config.ref;
    }
    if (hasValidKey(config)) {
      key = '' + config.key;
    }

    self = config.__self === undefined ? null : config;
    source = config.__source === undefined ? null : config;
    // Remaining properties are added to a new props
    for (propName in config) {
      if (hasOwnProperty.call(config, propName) &&
        props[propName] = config[propName];
    }
  }
}
```

```
https://unpkg.com/react-dom@16.5.2/umd/react-dom.development.js
```

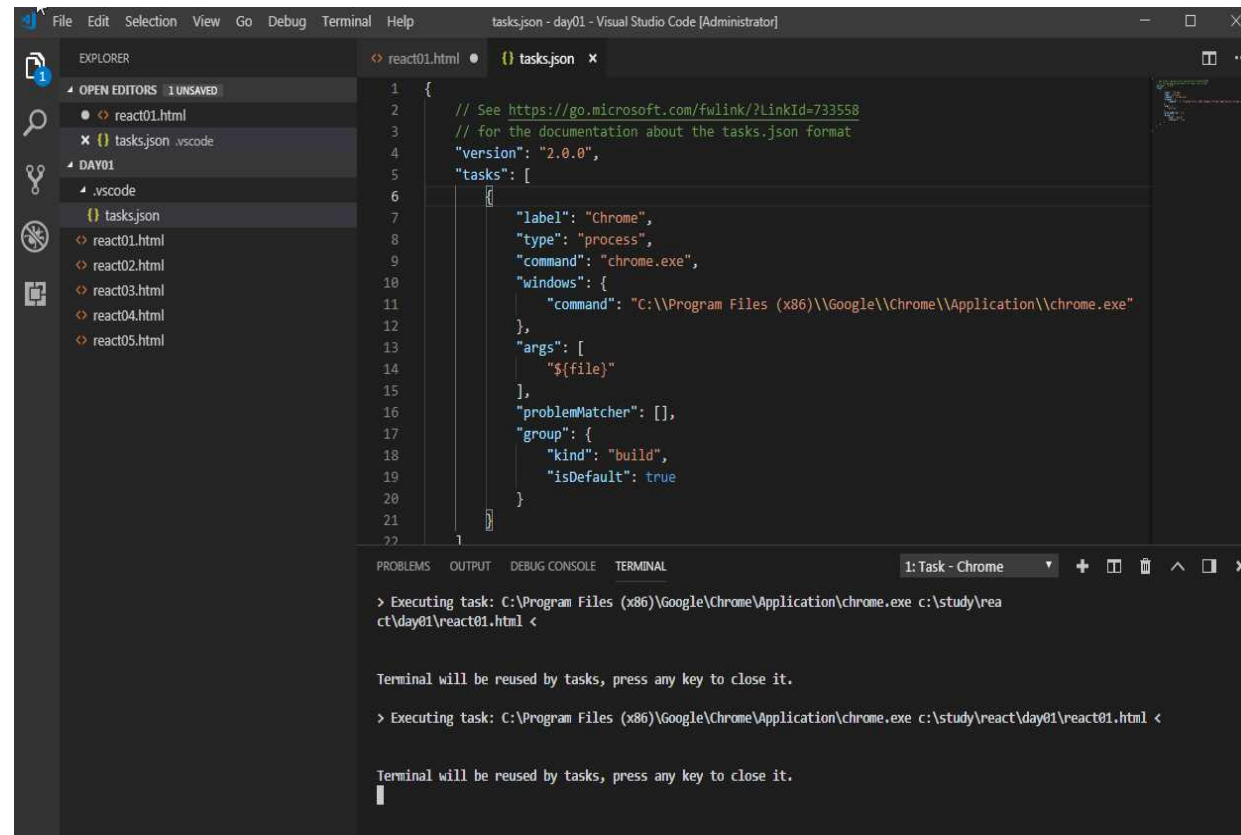
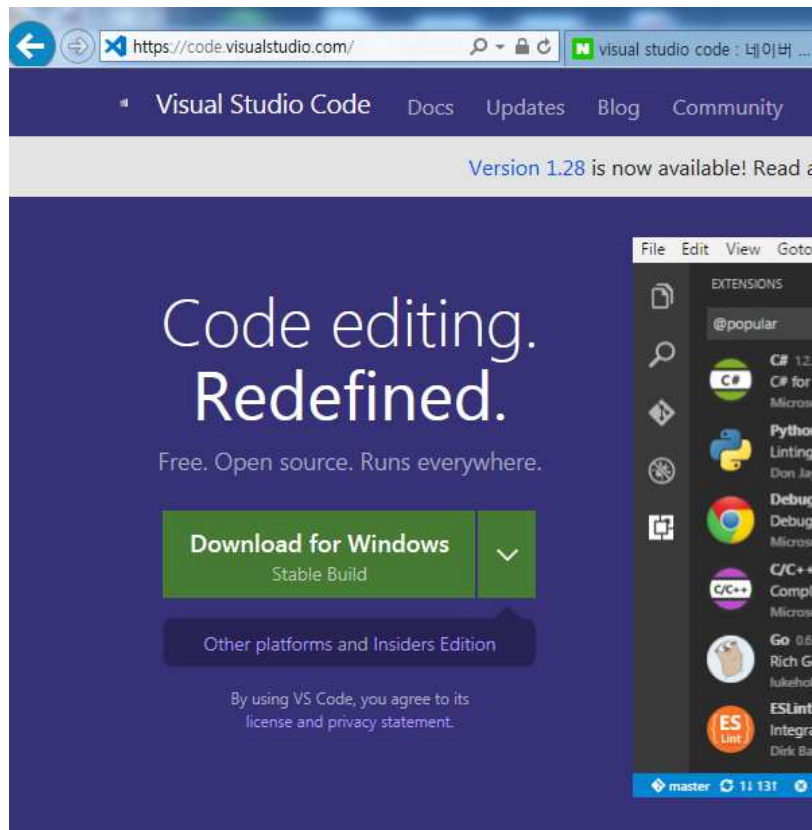
```
function createElement(type, props, rootContainerElement, parentNamespace) {
  var isCustomComponentTag = void 0;

  // We create tags in the namespace of their parent container, except HTML
  // tags get no namespace.
  var ownerDocument = getOwnerDocumentFromRootContainer(rootContainerElement);
  var domElement = void 0;
  var namespaceURI = parentNamespace;
  if (namespaceURI === HTML_NAMESPACE) {
    namespaceURI = getIntrinsicNamespace(type);
  }
  if (namespaceURI === HTML_NAMESPACE) {
    {
      isCustomComponentTag = isCustomComponent(type, props);
      // Should this check be gated by parent namespace? Not sure we want to
      // allow <SVG> or <math>.
      !(isCustomComponentTag || type === type.toLowerCase()) ? warning$1(false, '<svg /> is using incorrect lowercase for HTML elements.', type) : void 0;
    }

    if (type === 'script') {
      // Create the script via .innerHTML so its "parser-inserted" flag is
      // set to true and it does not execute
      var div = ownerDocument.createElement('div');
      div.innerHTML = '<script><' + '/script>'; // eslint-disable-line
      // This is guaranteed to yield a script element.
      var firstChild = div.firstChild;
      domElement = div.removeChild(firstChild);
    } else if (typeof props.is === 'string') {
      // $FlowIssue `createElement` should be updated for Web Components
      domElement = ownerDocument.createElement(type, { is: props.is });
    }
  }
}
```

6. 에디터 설치

- Visual Studio Editor 다운받아 기본 설치 하시면 됩니다.
- 무료인데 생각보다 속도도 빠르고 괜찮은것 같습니다.
- 단축키 컨트롤 K,O : 폴더열기 c:/study/react/day01



7. React Command 2개 – Hello React

- 첫번째 리액트 명령어 : ReactDOM.render(displayResult, root)
- 인자1은 나타낼결과 html 입니다.
- 인자2는 나타낼위치 id 입니다.
- 두번째 리액트 명령어 : React.createElement(html?태그, 속성, 보여줄컨텐츠)
- 인자1은 html태그 : h1, a 등등
- 인자2는 속성 : a 태그이면 href같은게 속성
- 인자3은 나타낼컨텐츠입니다.: Hello react !!

- 예제)

```
const displayResult = React.createElement(  
  "h1" ,  
  null ,  
  "Hello React !!!"  
)
```

- 한번 각자 노트북에 코딩해보시죠.

7. React Command 2개

– cheat sheet

–

```
<> react01.html ●
1  <html>
2
3  <head>
4      <script crossorigin src="https://unpkg.com/react@16/umd/react.development.js"></script>
5      <script crossorigin src="https://unpkg.com/react-dom@16/umd/react-dom.development.js"></script>
6
7  </head>
8  <div id="root"></div>
9
10 <script type="text/javascript">
11 //리액트 자바스크립트를 여기서 코딩하면 됩니다.
12
13 </script>
14
15 </html>
16
```

8. ES6 History

-1995년 Mocha로 시작 자바스크립트로 개명후 쪽쓰임

* java와 javascript는 이름만 비슷할뿐 거의 다르다고 할 수 있죠.
(코끼리와 바다코끼리정도 라네여..)

-1996년 표준화 작업을 일부 하면서 ecmascript라고도 불림 ES네요.

ES1 = ES1997

2001년 JSON 등장

2005년 AJAX

2006년 JQuery

2008년 크롬V8, NodeJS

ES6 = ES2015 CLASS, 화살표함수, let, 템플릿리터럴 , 매개변수에 레스트 스프레드

ES7 = ES2016

90년대 후반 2000년대 초반에 동적HTML형태로 많이 사용되다.. 관심도 하락
Html결가지로 조금씩 쓰는형태였다

NodeJS가 나오면서 서버에서 자바스크립트로 코딩이 되면서 다시 부각

왜 부각될까?

다양한 언어를 배우필요없이 자바스크립트만 알면 풀스택 개발이 가능해지므로
좋아하는 개발자가 많아지고 있네여.

9. 리액트코딩 기초 : 가독성 높게 JSX로 해봅시다.

- 리액트에서 자바스크립트와 Html을 섞어쓰는게 JSX
- 처음에보니 좀 이상하긴 합니다.
- html과 리액트 부분이 헷갈리기도 하고
- 근데 중첩되는 매우 복잡한 엘리먼트를 생성할때는 이게 당근 편하겠네요.
- `<h1> Hello React with JSX </h1>` 이것을 표현하려면 JSX로는 그대로 쓰면됩니다.
- 이렇게 쓰면 중간에서 트랜스파일링 하는거죠 바벨이란 분이
- 위에 라이브러리를 추가해 줍니다. 바벨
- 직접 코딩해보시죠.

```
1  <html>
2
3  <head>
4    <script crossorigin src="https://unpkg.com/react@16/umd/react.development.js"></script>
5    <script crossorigin src="https://unpkg.com/react-dom@16/umd/react-dom.development.js"></script>
6    <script crossorigin src="https://cdnjs.cloudflare.com/ajax/libs/babel-standalone/6.26.0/babel.js">
7
8  </head>
9  <div id="root"></div>
10
11  <script type="text/babel">
12
13    //가독성높게 JSX로 해봅시다.
14    const displayResult = <h1>Hello react !! </h1>
15
16    ReactDOM.render(displayResult, root)
17
18  </script>
19
20 </html>
```

9. 리액트코딩 기초 : JSX로 A 태그 만들며 연습 한번더

file:///C:/study/react/day01/react03.html



Hello react !! 홍길동

- 결과물

- 홍길동은 const로 받아서 표시해보죠
- 스타일은 가운데정렬에 색상을 입혀보죠 const로 받아서 처리
- 링크도 const로 받아서 연결해 봅니다.
- 리액트에서 const 받은것을 표시하는 것은 { } 사용합니다.
- `<div style={firstStyle}><a > Hello react !! {receivedName} </div>`

```
<script type="text/babel">

const receivedName = "홍길동";
const linkUrl = "http://www.naver.com"
const firstStyle = {
  color : "red",
  textAlign : "center"
}

const displayResult = <div style={firstStyle}><a href={linkUrl} style={firstStyle} > Hello react !! {receivedName} </a></div>

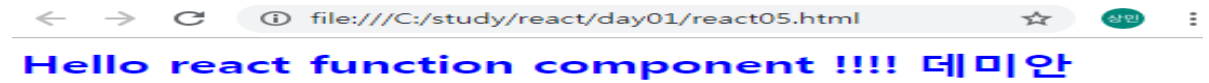
const root = document.getElementById("root")

ReactDOM.render(displayResult, root)

</script>
```

9. 리액트코딩 기초 : JSX로 사용자평션을 만들어 사용

– 결과물



-
- `const displayResult03 = <F01 userName="데미안" color="blue"></F01>`
- `<F01` 여기서부터 이제 무자게 헛갈리더라구여.
- 내가 만든 사용자정의 평션입니다.
- `<F01` 태그를 내가 정의하는 것이죠
- 속성을 리액트에서 간단히 전달이 쉽게 props라는 것으로 정의해논걸 사용합니다.
- 속성에 객체가 전달되면 `{{ }}` 이런식으로 사용가능합니다.

```
const displayResult03 = <F01 userName="데미안" color="blue"></F01>
ReactDOM.render(displayResult03, userFunc)

//평션으로 해봅시다. 평션에 속성까지 전달해서
function F01 (props) {
  |   return <h2 style={{color: props.color}} > Hello react function component !!!! {props.userName} </h2>
  }
}
```

9. 리액트코딩 기초 : JSX로 CLASS 만들어 사용

– 결과물



- `const displayResult03 = <F01 userName="데미안" color="blue"></F01>`
- `<F01` 여기서부터 이제 무자게 헛갈리더라구여.
- 내가 만든 사용자정의 평션입니다.
- `<F01` 태그를 내가 정의하는 것이죠
- 속성을 리액트에서 간단히 전달이 쉽게 props라는 것으로 정의해논걸 사용합니다.
- 속성에 객체가 전달되면 `{{ }}` 이런식으로 사용가능합니다.

```
const displayResult03 = <F01 userName="데미안" color="blue"></F01>
ReactDOM.render(displayResult03, userFunc)

//평션으로 해봅시다. 평션에 속성까지 전달해서
function F01 (props) {
  |   return <h2 style={{color: props.color}} > Hello react function component !!!! {props.userName} </h2>
  }
}
```