

2주차 이론

2024년 3월 15일 금요일 오후 2:14

HTML(Hyper Text Markup Language) : 웹사이트의 모습을 기술하기 위한 마크업 언어

Hyper Text

- 참조(하이퍼링크)를 통해 독자가 한 문서에서 다른 문서로 즉시 접근할 수 있는 텍스트

Markup Language

- 태그 등을 이용하여 문서나 데이터의 구조를 명기하는 언어
- 문서가 화면에 표시되는 형식을 나타내거나 데이터의 논리적인 구조를 명시하기 위한 규칙들을 정의한 언어의 일종

HTML 문서구조

```
<!doctype html>
<html>
  <head>
    타이틀, 메타태그, 스타일시트 영역
  </head>
  <body>
    본문 영역
    <script>
      자바스크립트 영역
    </script>
  </body>
</html>
```

CSS

- Style sheet 언어
- HTML 문서에 있는 요소들에 선택적으로 스타일을 적용
- SASS : CSS의 단점을 보완하기 위해 만든 CSS 전처리기
- HTML 문서의 색이나 모양 등 외관을 꾸미는 언어
- CSS로 작성된 코드를 style sheet라고함

스타일

- 문서를 장식하거나 정해진 양식에 맞게 내용을 다듬는 역할
- 문서 작성 프로그램에서 작성하는 모든 문서는 스타일을 적용함

스타일 시트

- 문서의 스타일을 정의
- 화면에 출력하기 위한 외형적인 모양과 형식에 관한 스타일 규칙들의 모임

- 스타일 시트에 의해 웹 문서가 브라우저에 어떻게 보여지는지가 결정

CSS기능

- 색상과 배경, 텍스트, 폰트, 박스 모델, 비주얼 포맷 및 효과
- 리스트, 테이블, 사용자 인터페이스

Javascript

- 스크립트 언어
- 웹 페이지가 동작하는 것을 담당
- Ecma international의 프로토타입 기반의 프로그래밍 언어

DOM

HTML DOM

- 웹 페이지에 작성된 HTML 태그 당 객체 생성

목적

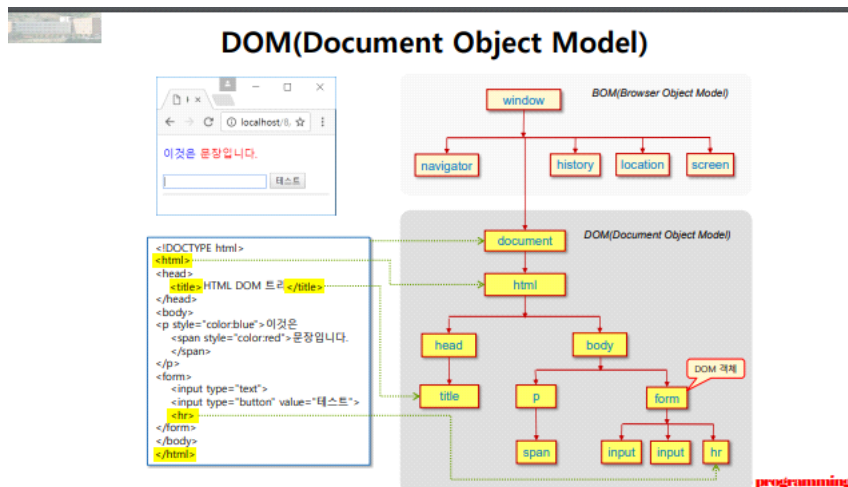
- HTML 태그가 출력된 모양이나 콘텐츠를 제어하기 위해
- DOM객체를 통해 각 태그의 CSS3 스타일 시트 접근 및 변경
- HTML 태그에 의해 출력된 텍스트나 이미지 변경

DOM 트리

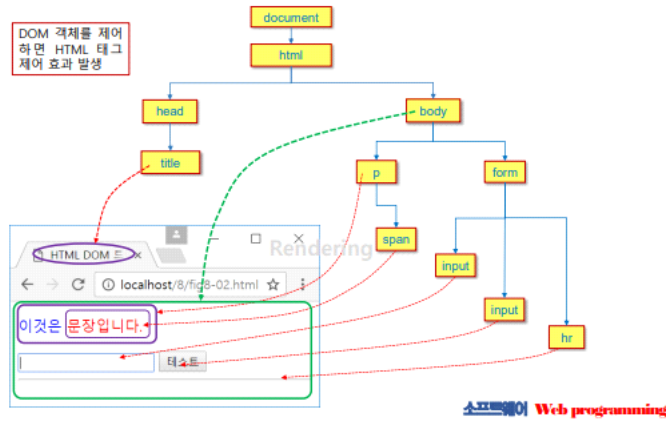
- HTML 태그의 포함관계에 따라 DOM객체의 트리 생성
- DOM 트리는 부모 자식 관계

DOM 객체

- HTML 태그 당 하나의 DOM객체 생성
- -- DOM노드, DOM엘리먼트 라고도 불림



DOM(Document Object Model)

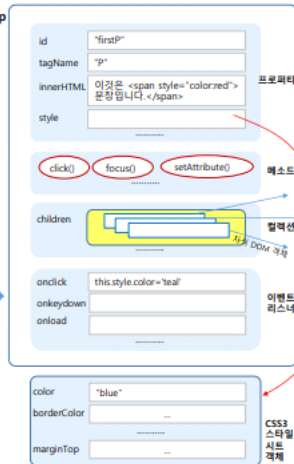


DOM 객체 p

DOM 객체의 구성

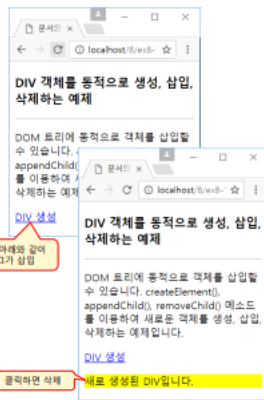
- 프로퍼티(property)
- 메소드(method)
- 컬렉션(collection)
- 이벤트 리스너(event listener)
- CSS3 스타일

```
<p id="firstP" style="color:blue" onclick="this.style.color='teal'">
  이것은
  <span style="color:red">
    문장입니다.
  </span>
</p>
```



DOM(Document Object Model)

```
<!DOCTYPE html>
<html>
<head> <title>문서의 동적 구성</title>
<script>
function createDIV() {
  var obj = document.getElementById("parent");
  var newDIV = document.createElement("div");
  newDIV.innerHTML = "새로 생성된 DIV입니다.";
  newDIV.setAttribute("id", "myDIV");
  newDIV.style.backgroundColor = "yellow";
  newDIV.onclick = function() {
    var p = this.parentElement; // 부모 HTML 태그 요소
    p.removeChild(this); // 자신을 부모로부터 제거
  };
  obj.appendChild(newDIV);
}
</script>
</head>
<body id="parent">
<h3>DIV 객체를 동적으로 생성, 삽입, 삭제하는 예제</h3>
<hr>
<p>DOM 트리에 동적으로 객체를 삽입할 수 있습니다.
createElement(), appendChild(), removeChild() 메소드를 이용하여 새로운 객체를 생성,
삽입, 삭제하는 예제입니다.</p>
<a href="javascript:createDIV()">DIV 생성</a><p>
<p>
</body>
</html>
```

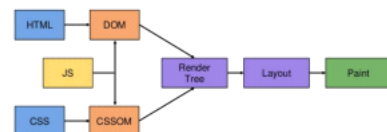


Rendering

Rendering

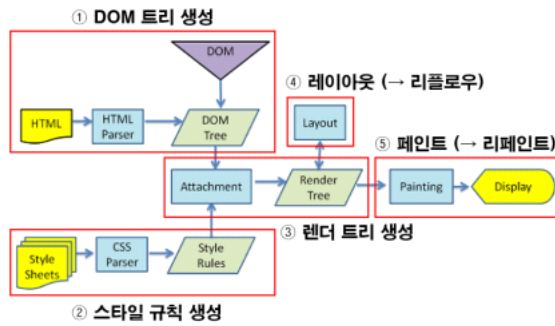
- 실시간으로 웹사이트가 그려지는 과정
 - HTML을 파싱하여 DOM 트리를 만든다.
 - CSS를 파싱하여 CSSOM 트리를 만든다.
 - DOM과 CSSOM을 결합하여 렌더링 트리를 만든다.
 - 렌더링 트리에서 각 노드의 크기와 위치를 계산한다.
 - 개별 노드를 화면에 그린다.

Critical Rendering Path (CRP)





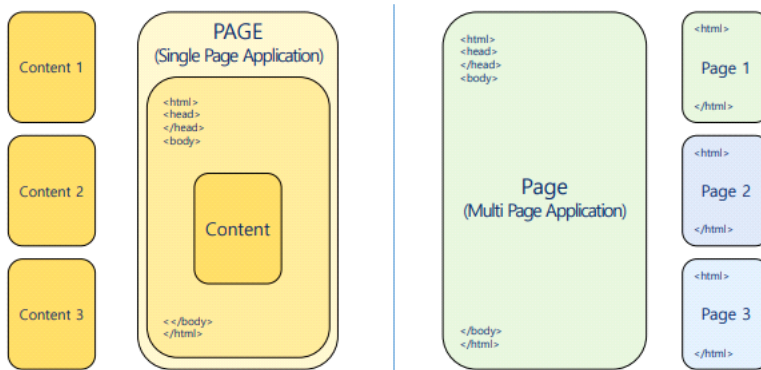
Rendering



소프트웨어 Web programming



SPA vs MPA



소프트웨어 Web programming

SPA

- 한 개의 page로 구성된 애플리케이션
- 모든 정적 리소스를 최초 한번에 다운로드
- 새로운 페이지 요청이 있을 때, 필요 데이터만 받아 갱신
- 서버로부터 완전한 새로운 페이지를 불러오지 않고 현재의 페이지를 동적으로 다시 작성함으로써 사용자와 소통하는 웹사이트
- CSR방식으로 렌더링함.

MPA

- 여러 개의 page로 구성된 애플리케이션
- 새로운 페이지를 요청할 때, 서버에서 렌더링 된 정적리소스(HTML,CSS,JS)가 다운로드
- 페이지를 이동하거나 새로고침하면, 전체페이지를 다시 렌더링함.
- SSR방식으로 렌더링.

SPA장점

- 전체 페이지를 업데이트 할 필요가 없기때문에, 빠르고 깜박거림이 없음
- 반응형 디자인으로 페이지 상의 무언가를 클릭할 시, 기다림 없이 즉각적인 반응
- 서버에게 정적 리소스를 최초 한번만 요청함/받은 데이터는 캐시로 전부 저장
- 모든 로컬 스토리지를 효과적으로 캐시할 수 있음/오프라인으로 사용가능
- 모바일 앱 개발을 염두에 둔다면, 동일한api사용하도록 설계가능
- 전 페이지를 다시 렌더링하지 않고 변경되는 부분만 갱신하므로 새로고침이 발생하지 않아, 네이티브 앱과 유사한 사용자경험을 줄 수 있다.

- 모바일개발 시, 동일한 백엔드 코드를 재사용하게 할 수 있다.
- 서버의 템플릿 연산을 클라이언트로 분산이 가능
- 컴포넌트 별 개발에 용이하고, 재사용 가능

SPA단점

- 최초 로딩 시 한번에 받아오기 때문에, 초기 페이지 로딩시간이 길
- JS로 구축된다.(JS를 읽지 못하는 검색엔진에선 색인이 되지 않는 문제가 발생가능)
- SSR에서는 서버에서 세션으로 관리, 하지만 CSR방식에선 쿠키에 저장해야 하며, 보안이 취약

MPA장점

- 완성된 형태의 HTML파일을 서버로부터 전달 받는다.
- 서버에서 미리 렌더링 후 가져오기 때문에, 첫 로딩이 굉장히 짧음
- 다중 페이지로 원하는 만큼 페이지 추가 가능
- 여러 페이지를 생성할 수 있기 때문에, 훨씬 수많은 키워드를 타겟팅 할 수 있다.
- 여러 HTML파일을 다운로드하여, 페이지를 크롤링하기에, 검색엔진이 작동하는 방식에 더 적합함.
- 유저가 보는 화면 방식이 웹 크롤러가 보는 방식과 같아 SEO에 더 적합함.

MPA단점

- 새 페이지 이동 시, 깜박인다.
- 요청이 들어올 때마다 페이지를 새로 패치해야 하기에, 로딩시간이 발생
- 페이지 이동 시, 불필요한 템플릿도 중복해서 로딩함.
- 클라이언트가 JS 파일을 모두 다운로드하고 적용하기 전까지 각 기능은 동작X
- 프론트와 백의 구분이 확연하지 않아, 서버 쪽과 클라이언트 쪽 언어 모두 필요
- 지속적 유지보수가 힘들
- 애플리케이션 개발 속도가 길어짐



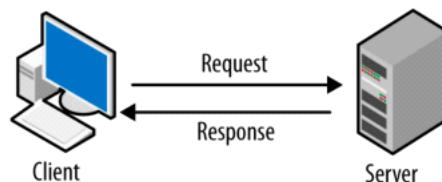
Client - Server

Client

- 특정 서비스를 이용하는 사용자(고객)
- 웹 브라우저

Server

- 특정 서비스를 제공하는 자



Browser

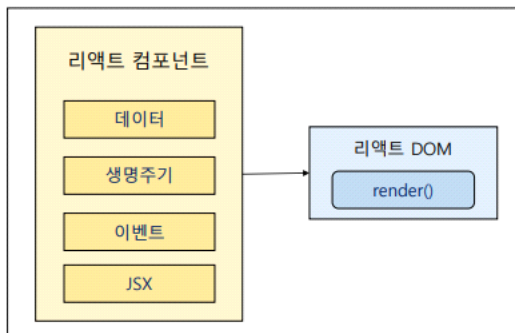
Web browser

- HTML 문서와 그림, 멀티미디어 파일 등 월드 와이드 웹을 기반으로 한 인터넷의 콘텐츠에 접근하기 위한 응용 프로그램
- 웹 브라우저, 인터넷 브라우저 또는 웹 탐색기는 웹 서버에서 이동하며(navigate) 쌍방향으로 통신하고 HTML 문서나 파일을 출력하는 그래픽 사용자 인터페이스 기반의 응용 소프트웨어
- 대표적인 HTTP 사용자 에이전트



React

The library for web and native user interfaces



React



React

The library for web and native user interfaces

리액트란?

- 사용자 인터페이스를 만들기 위한 JavaScript 라이브러리
- 사용자와 웹 사이트의 상호작용을 돕는 인터페이스를 만들기 위한 자바스크립트 기능 모음집
- 2013년에 페이스북에서 발표한 오픈소스 자바스크립트 프레임워크
- 가상 DOM(Document object model)과 JSX(JavaScript XML) 방식으로 동작하는 프레임워크
- SPA(Single Page Application) 개발을 위한 프레임워크
- [사용자 인터페이스를 구축하기 위한 선언적이고 효율적이며 유연한 JavaScript 라이브러리](#)
- "컴포넌트"라고 불리는 작고 고립된 코드의 파편을 이용하여 복잡한 UI를 구성

소프트웨어 Web programming

React 장점

- 완성도가 높고 이해하기 쉬운 개발 워크 플로우
- 유연성과 호환성이 좋음
- 컴포넌트 재사용성이 쉬움

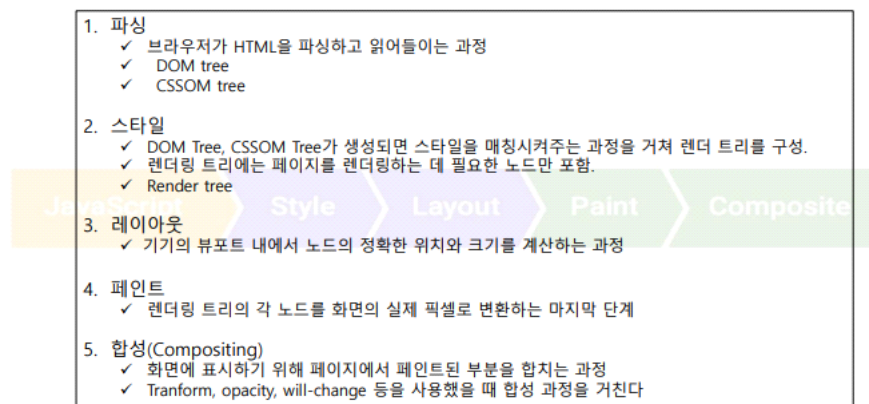
- 다양한 툴 제공
- React native의 강력한 기능
- 시장에 영향을 미치는 거대한 커뮤니티와 리소스
- HTML 확장을 위한 JSX 구문
- React hook

단점

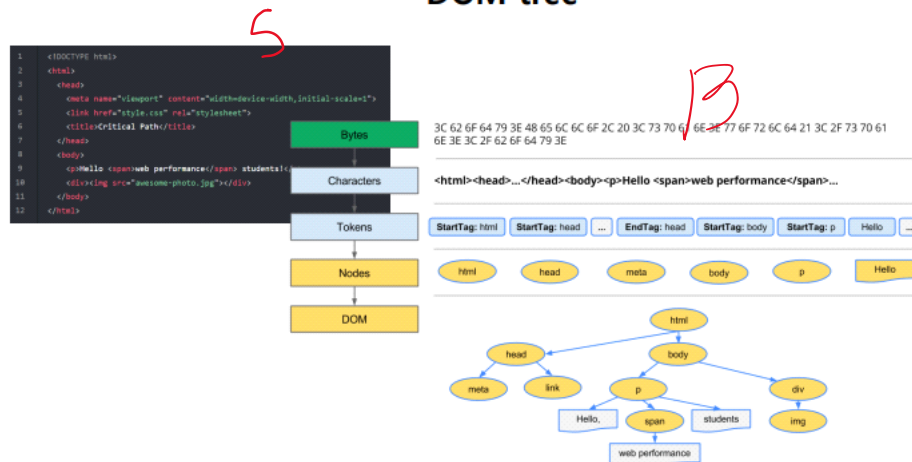
- 방대한 학습량
- 버전업
- 높은 상태 관리 복잡도



브라우저 로딩 과정

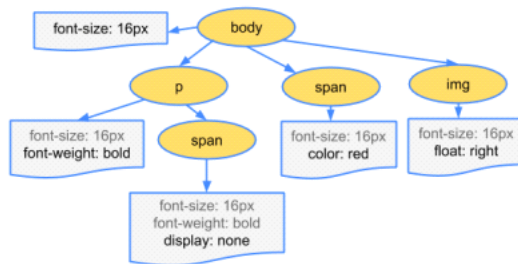


DOM tree

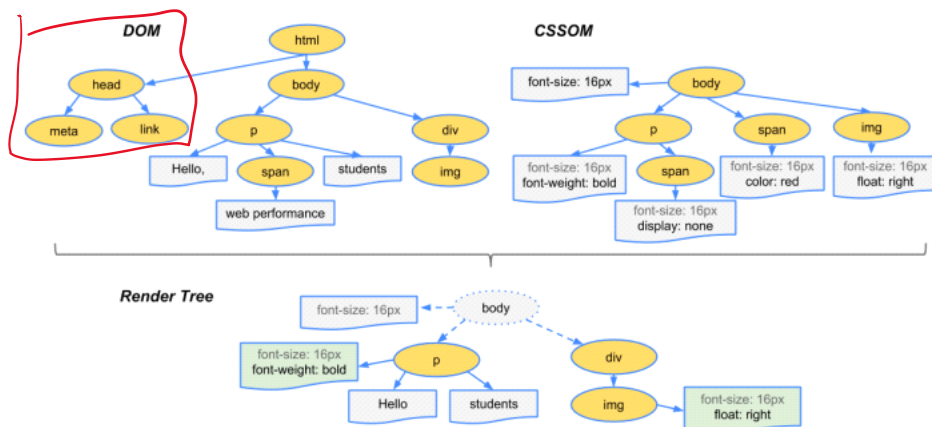


CSSOM tree *style*

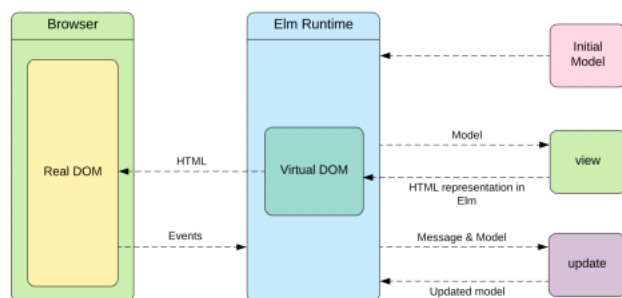
```
1 body { font-size: 16px }
2 p { font-weight: bold }
3 span { color: red }
4 p span { display: none }
5 img { float: right }
```



Render tree



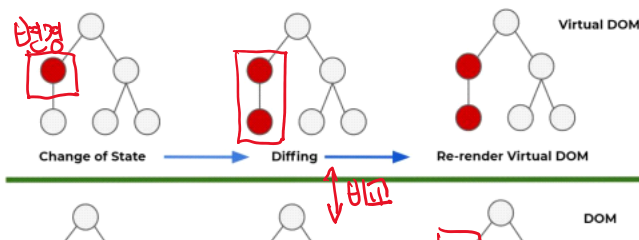
Virtual DOM ← Real DOM을 복제

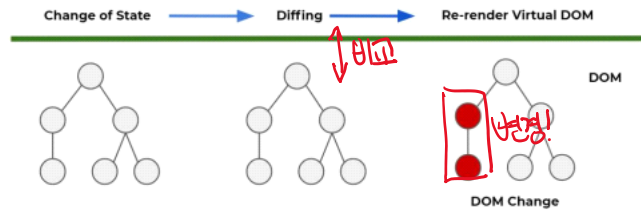


DOM에 변경이 있을 경우 렌더트리를 재생성하고(모든 요소들의 스타일이 다시 계산됨) 레이아웃을 만들고 페인팅을 하는 과정이 다시 반복되는 것을 알 수 있었다. 즉, 브라우저가 연산을 많이 해야한다는 이야기이며, 전체적인 프로세스를 비효율적으로 만든다는 것이다.

그래서 등장한 개념이 가상돔(VirtualDOM) 이다.

Virtual DOM





뷰(HTML)에 변화가 있을 때, 구 가상돔(Old Node)과 새 가상돔(New Node)을 비교하여 변경된 내용만 DOM에 적용한다.
 이를 통해 브라우저 내에서 발생하는 연산의 양(정확히는 렌더링 과정)을 줄이면서 성능이 개선되는 것이다.

SPA 구조

Real DOM vs Virtual DOM

```

1 <div id="app">
2   <ul>
3     <li>
4       <input type="checkbox" class="toggle" />
5       todo list item 1
6       <button class="remove">삭제</button>
7     </li>
8     <li class="completed">
9       <input type="checkbox" class="toggle" checked="" />
10      todo list item 2
11      <button class="remove">삭제</button>
12    </li>
13  </ul>
14  <form>
15    <input type="text" />
16    <button type="submit">추가</button>
17  </form>
18 </div>
  
```

Real DOM

```

1 function virtualDom(type, props, ...children) {
2   return { type, props, children: children.flat() }
3 }
4
5 virtualDom('div', { id: 'app' },
6   virtualDom('ul', null,
7     virtualDom('li', null,
8       virtualDom('input', { type: 'checkbox', className: 'toggle' }),
9       'todo list item 1',
10      virtualDom('button', { className: 'remove' }, '삭제')
11     ),
12     virtualDom('li', { className: 'completed' },
13       virtualDom('input', { type: 'checkbox', className: 'toggle', checked: true }),
14       'todo list item 2',
15       virtualDom('button', { className: 'remove' }, '삭제')
16     ),
17   ),
18   virtualDom('form',
19     virtualDom('input', { type: 'text' }),
20     virtualDom('button', { type: 'submit' }, '추가'),
21   ),
22 );
  
```

Virtual DOM

보통 virtualDom 대신 h 로 표현

Babel

JavaScript에 컴파일러가 필요한 이유

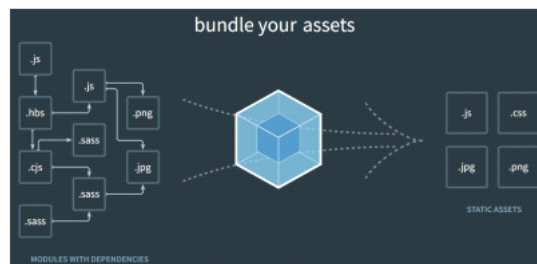
모든 브라우저가 최신 문법, 기술(ES6)을 지원하지 않기 때문에 구 기능(ES5)으로 변환하는 작업이 필요

Babel이 하는 일

- Transform syntax (구문 변환)
 - 트랜스파일링은 최신의 자바스크립트 문법을 오래된 브라우저가 이해할 수 있도록 오래된 문법으로 변환해 준다.
- babel-polyfill을 통해서 폴리필 기능을 지원
 - 폴리필은 오래된 브라우저에 네이티브로 지원하지 않는 사용자가 사용하는 메서드, 속성, API가 존재하지 않을 때 추가해 준다.
 - 바벨은 최신 문법을 오래된 문법으로 변환해 주는 트랜스파일러 역할만 할 뿐 최신 함수를 사용할 수 있는 건 아니다.
 - 폴리필은 프로그램이 처음 시작할 때, 지원하지 않는 기능들을 지원해 준다.
 - 바벨은 컴파일 때 실행되고 폴리필은 런타임에 실행된다.
- JSX and React
 - 바벨은 JSX 문법을 변환한다.

Babel is a JavaScript compiler.
 Use next generation JavaScript, today.

Webpack



웹팩(Webpack) 기본 사용법 (CLI)
 웹팩(Webpack) 기본 설정법 (Entry/Output/Loader/Plugins)
 웹팩(Webpack) 개발 편의 기능

Webpack

자바스크립트로 만든 프로그램을 배포하기 좋은 형태로 묶어주는 도구

모듈

프로그래밍 관점에서, 특정 기능을 갖는 작은 코드 단위

번들링

js, .css, .png ... 각각의 파일들을 모두 모듈로 보고, 이 모듈들을 배포용으로 하나의 파일로 통합시켜주는 작업



Index.html, index.js, App.js

index.html

- public 폴더에 있다.
- 메인 프로그램인 index.js에 대응되는 것으로, HTML 템플릿 파일이다.
- 이 파일이 직접 표시되는 것은 아니고, index.js에 의해 일어 와서 렌더링된 결과가 표시된다.
(index.html 이름을 바꿀 시 오류발생)

index.js

- src 폴더에 포함되어 있다. 메인 프로그램이라고 할 수 있다.
- 여기에서 HTML 템플릿 및 JavaScript의 컴포넌트를 조합하여 렌더링하고 실제 표시한다.

App.js

- src 폴더에 있다.
- 이것은 컴포넌트를 정의하는 프로그램이다.
- 실제로 화면에 표시되는 내용 등은 여기에서 정의된다.



Index.html, index.js, App.js

