

Web Programming

React programming



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.

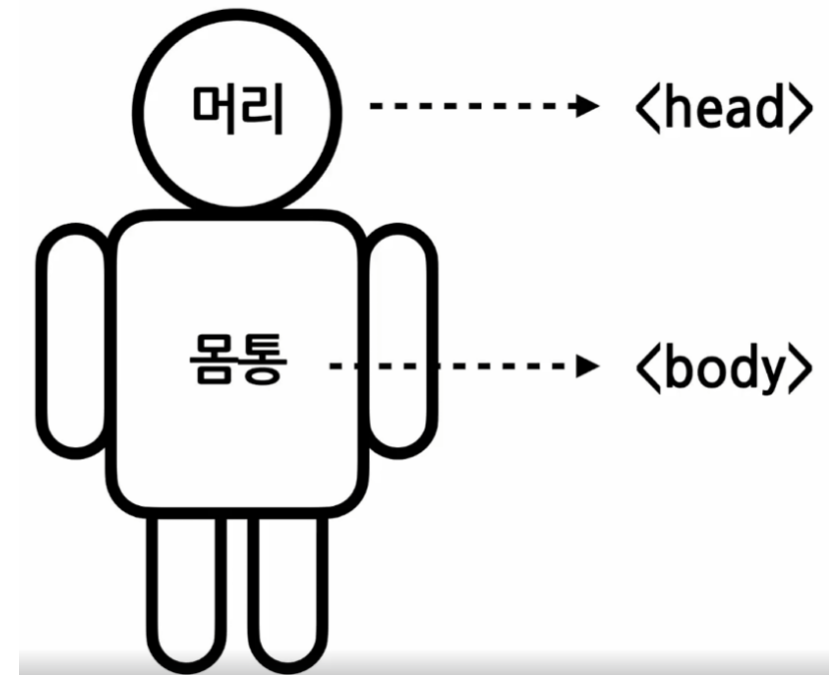
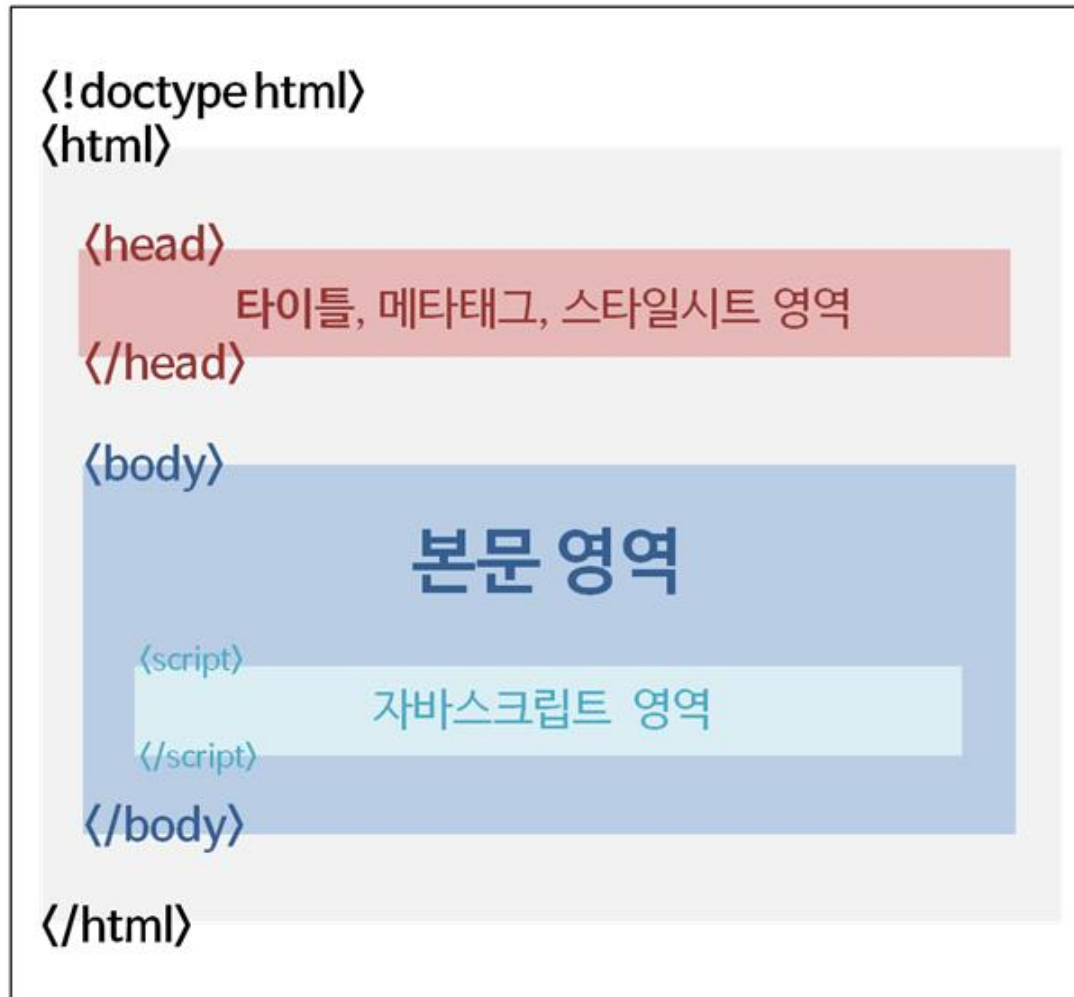
Markup Language

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





HTML 문서구조





CSS(Cascading Style Sheet)



```
body {  
  font: x-small  
  background: #  
  color: black;  
  margin: 0;  
  padding: 0;
```

CSS

- Style sheet 언어
- HTML 문서에 있는 요소들에 선택적으로 스타일을 적용



sass

- CSS의 단점을 보완하기 위해 만든 CSS 전처리기



CSS(Cascading Style Sheet)

스타일(style)

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

스타일시트(stylesheet)

- 문서의 스타일을 정의
- 화면에 출력하기 위한 외형적인 모양과 형식에 관한 스타일 규칙들의 모임
- 스타일시트에 의해 웹 문서가 브라우저에 어떻게 보여지는지가 결정됨

CSS(Cascading Style Sheet)

- HTML 문서의 색이나 모양 등 외관을 꾸미는 언어
- CSS로 작성된 코드를 스타일 시트(style sheet)라고 부름
- 현재 CSS3 : CSS level 3
- CSS1 -> CSS2 -> CSS3 -> CSS4(현재 표준화 작업 중)

CSS 기능

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





ECMAScript

What's the difference between JavaScript and ECMAScript?

ES6 and **JS**
ECMAScript 6 Javascript

Javascript

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



DOM(Document Object Model)

HTML DOM(간단히 DOM)

- 웹 페이지에 작성된 HTML 태그 당 객체(DOM 객체) 생성

목적

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

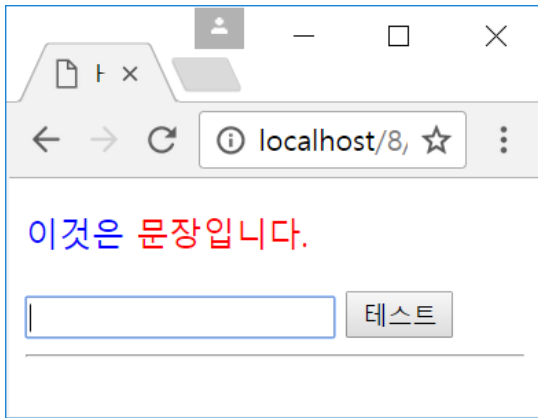
DOM 트리

- HTML 태그의 포함 관계에 따라 DOM 객체의 트리(tree) 생성
- DOM 트리는 부모 자식 관계

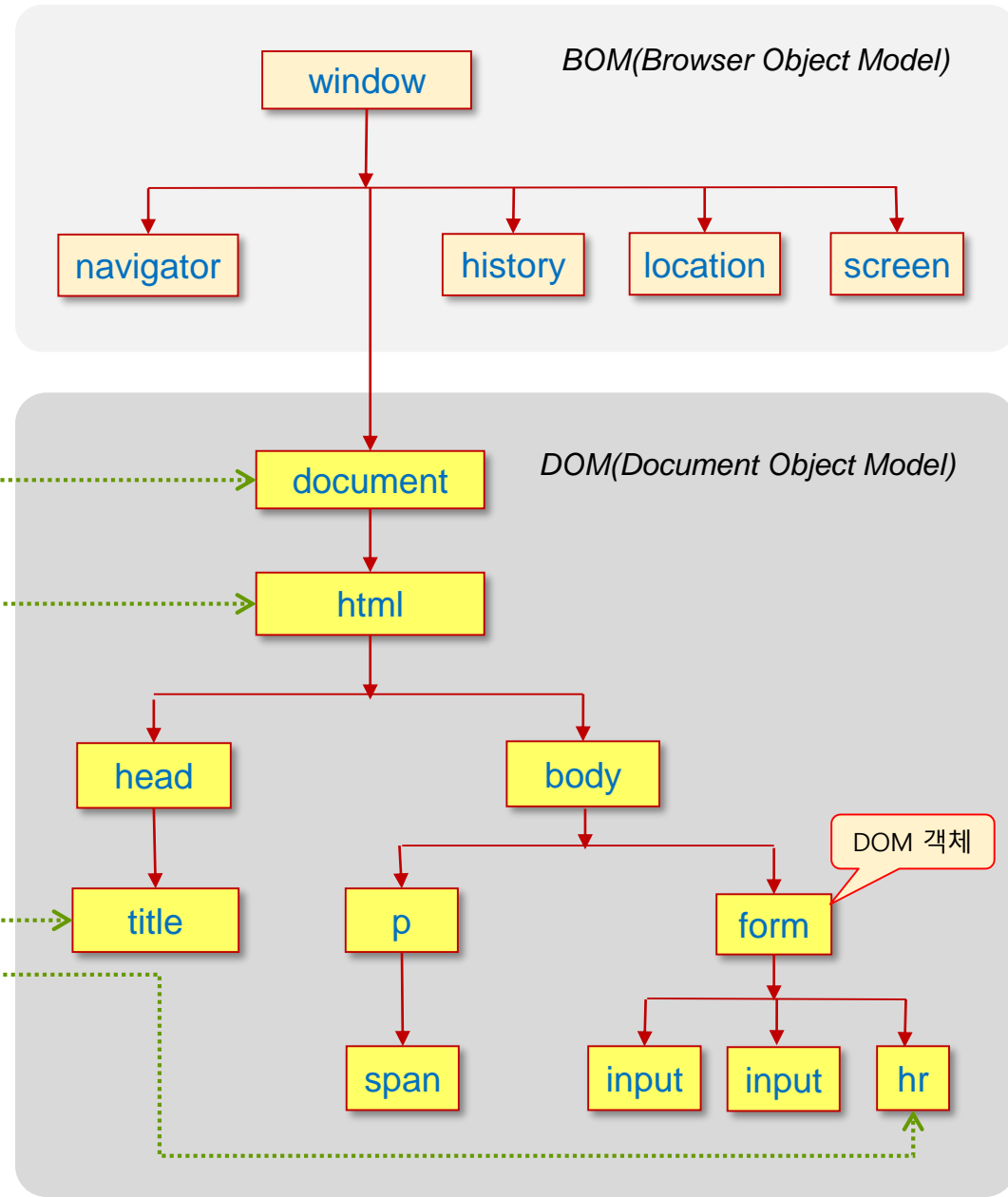
DOM 객체

- HTML 태그 당 하나의 DOM 객체 생성
 - DOM 노드(Node), DOM 엘리먼트(Element) 라고도 불림

DOM(Document Object Model)

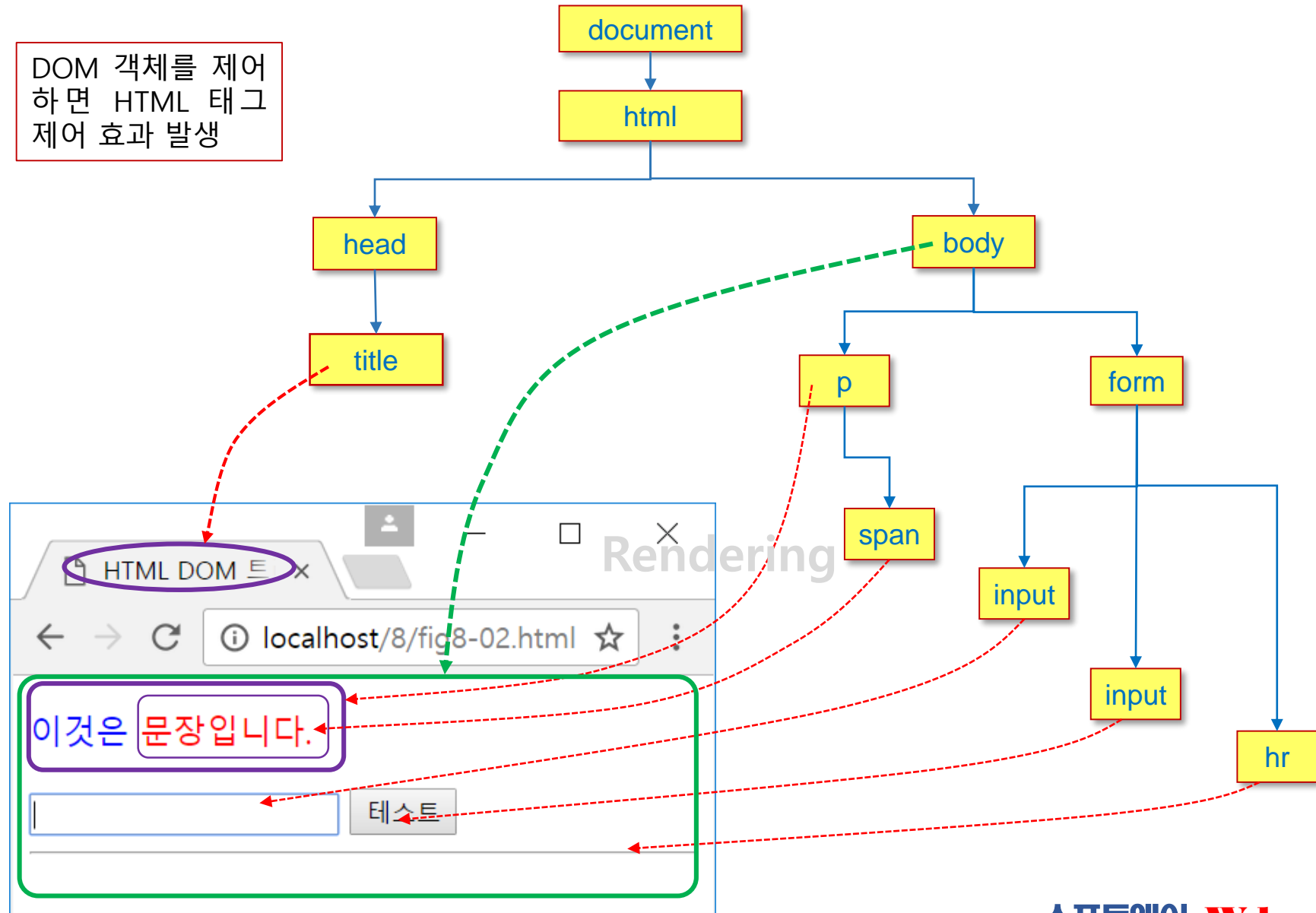


```
<!DOCTYPE html>
<html>
<head>
  <title> HTML DOM 트리 </title>
</head>
<body>
  <p style="color:blue"> 이것은
    <span style="color:red"> 문장입니다.
  </span>
</p>
  <form>
    <input type="text">
    <input type="button" value="테스트">
    <hr>
  </form>
</body>
</html>
```



DOM(Document Object Model)

DOM 객체를 제어
하면 HTML 태그
제어 효과 발생



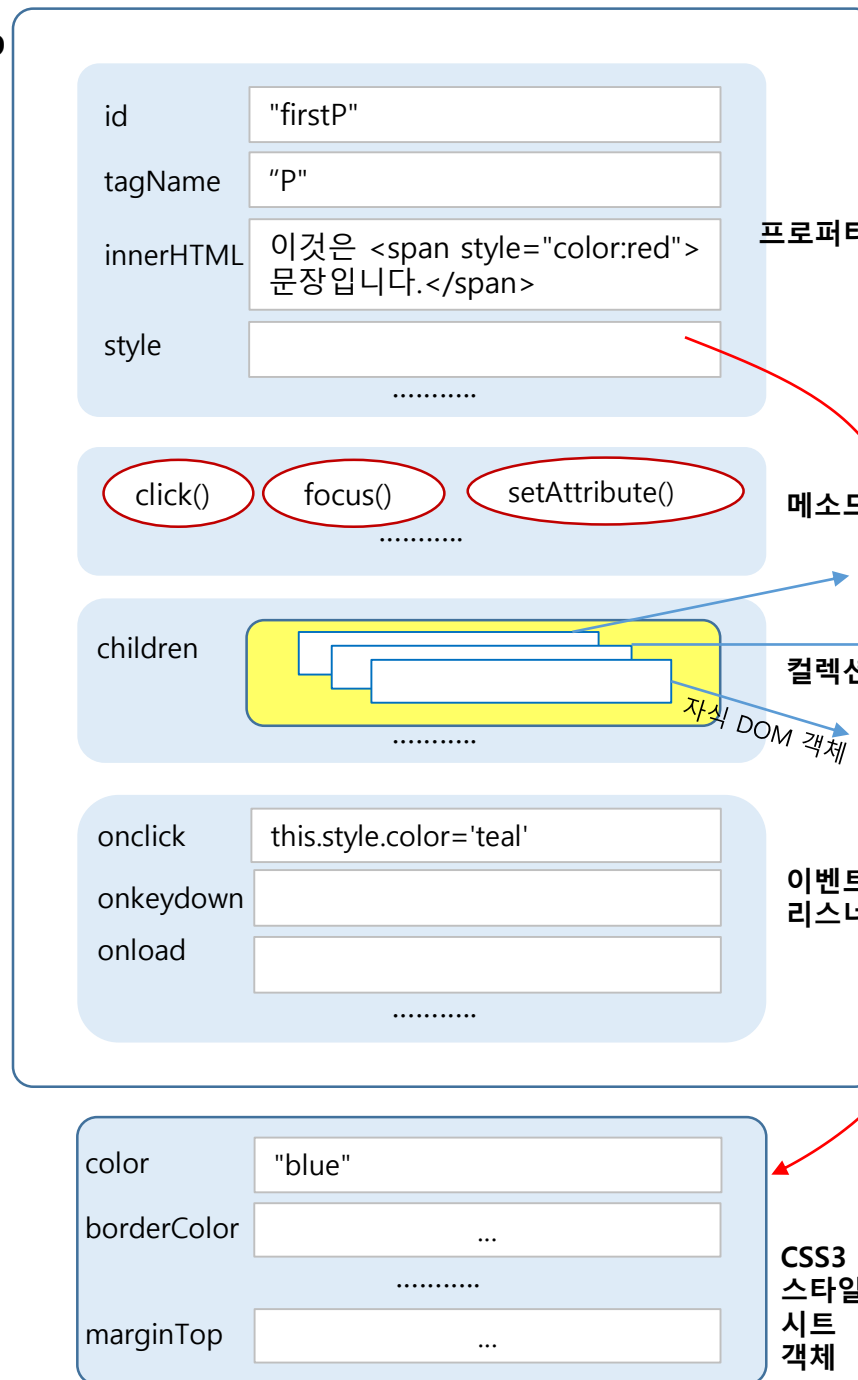


DOM 객체 p

DOM 객체의 구성

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

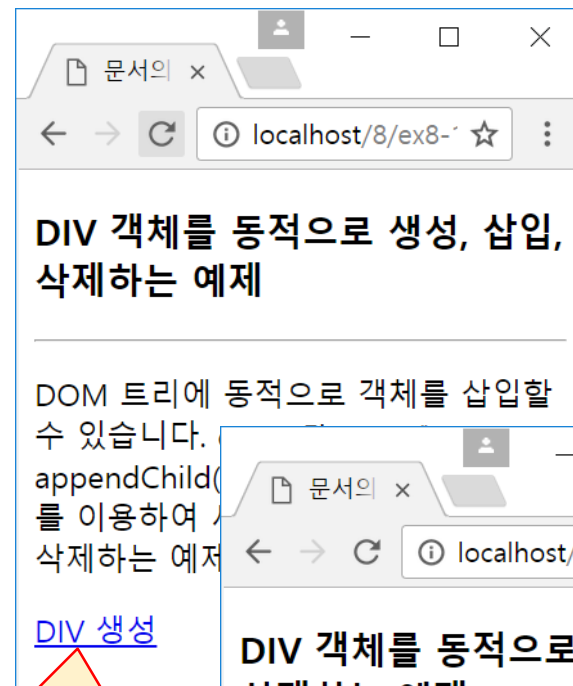
```
<p id="firstP"  
  style="color:blue"  
  onclick="this.style.color='teal'">  
  이것은  
  <span style="color:red">  
    문장입니다.  
  </span>  
</p>
```



DOM(Document Object Model)

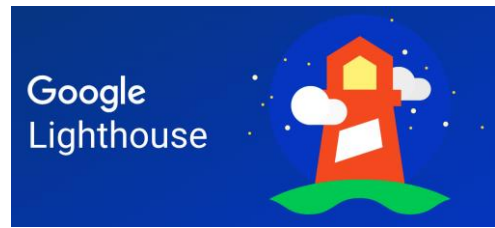
```
<!DOCTYPE html>
<html>
<head> <title>문서의 동적 구성</title>
<script>
function createDIV() {
  var obj = document.getElementById("parent");
  var newDIV = document.createElement("div");
  newDIV.innerHTML = "새로 생성된 DIV입니다.";
  newDIV.setAttribute("id", "myDiv");
  newDIV.style.backgroundColor = "yellow";
  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>
```

클릭하면 아래와 같이
<div> 태그가 삽입





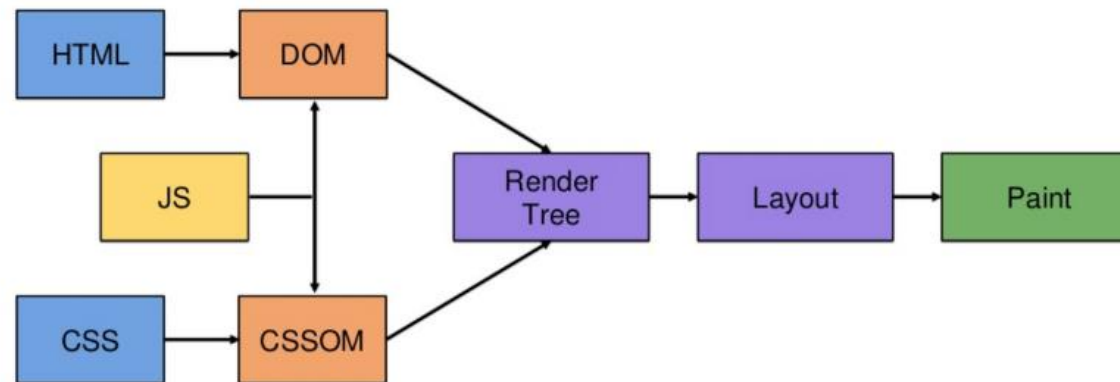
Rendering



Rendering

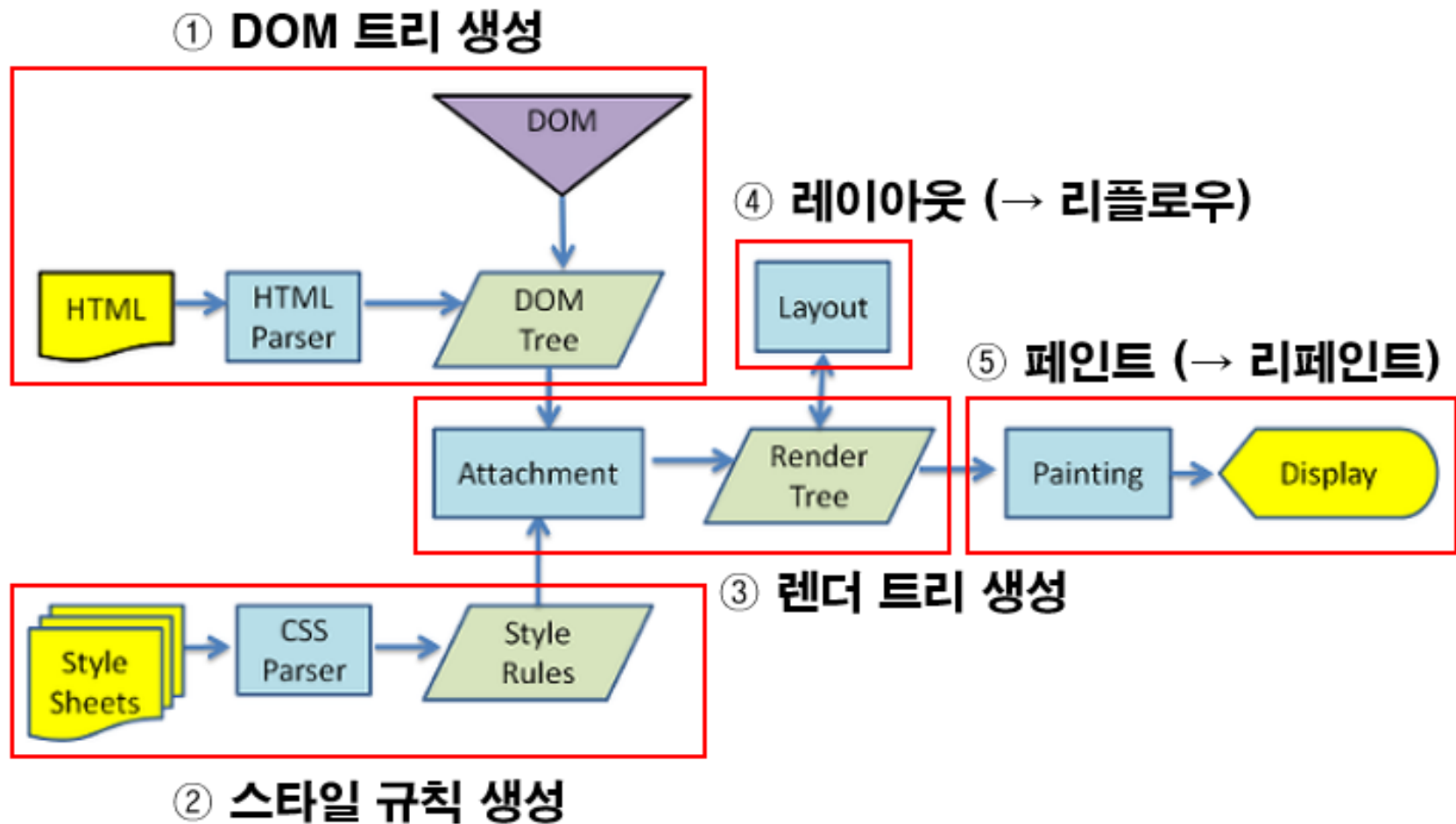
- 실시간으로 웹사이트가 그려지는 과정
 - HTML을 파싱하여 DOM 트리를 만든다.
 - CSS를 파싱하여 CSSOM 트리를 만든다.
 - DOM 과 CSSOM 을 결합하여 렌더링 트리를 만든다.
 - 렌더링 트리에서 각 노드의 크기와 위치를 계산한다.
 - 개별 노드를 화면에 그린다.

Critical Rendering Path (CRP)



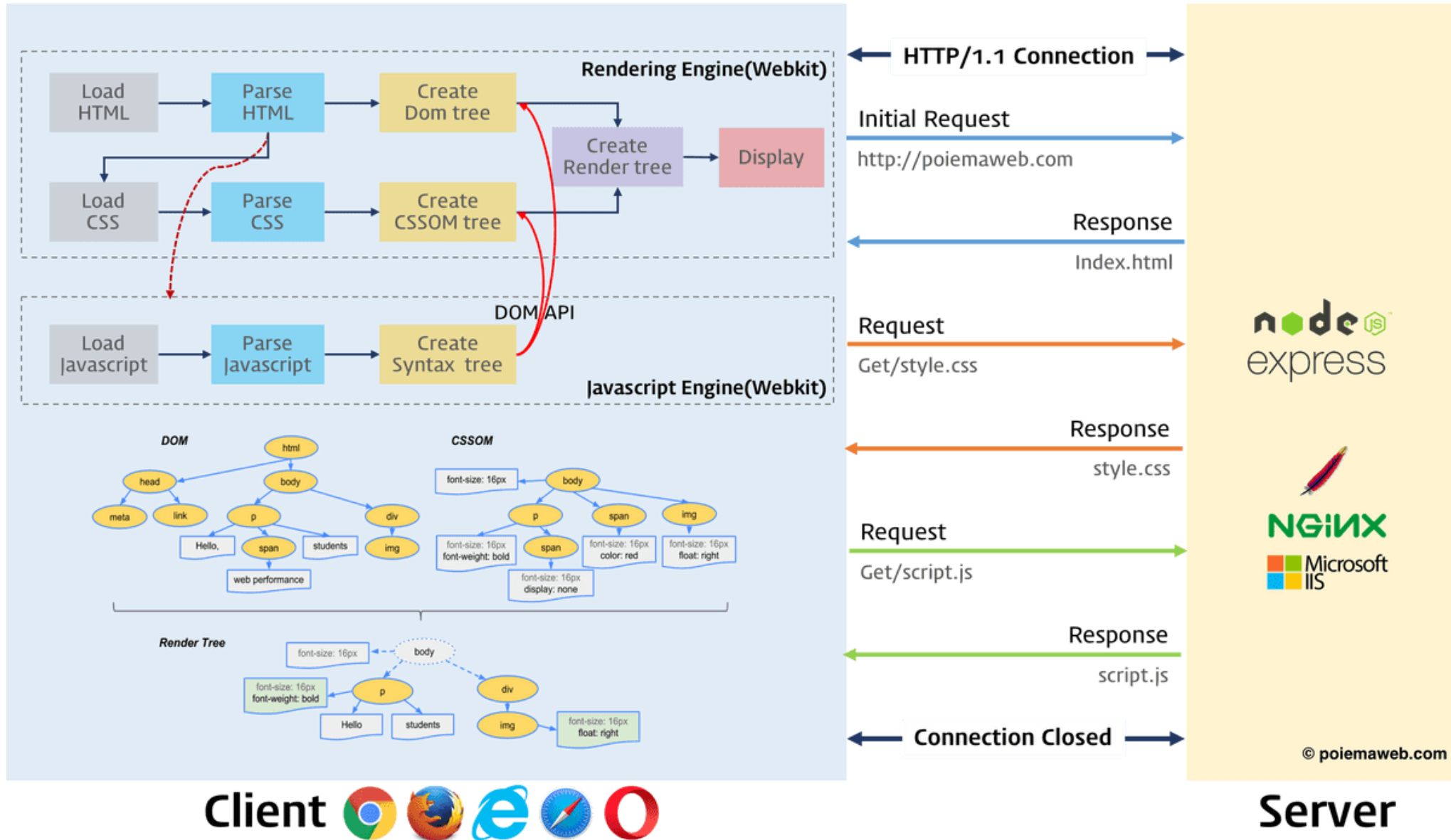


Rendering



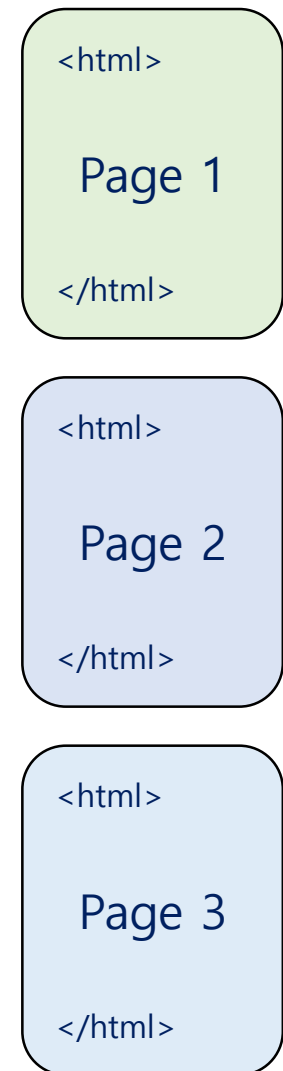
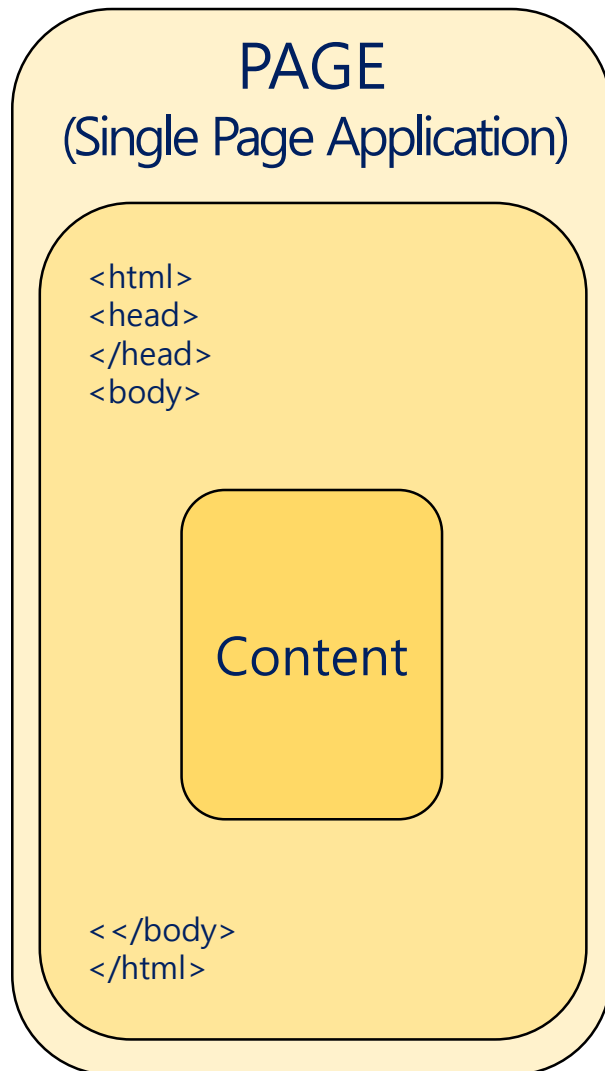
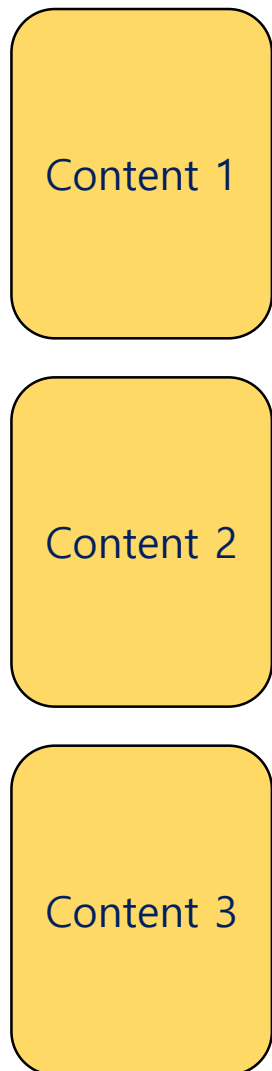


Rendering





SPA vs MPA

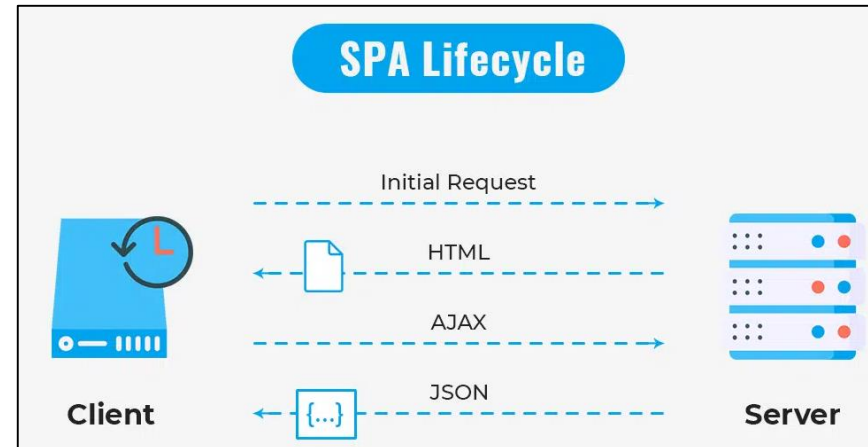
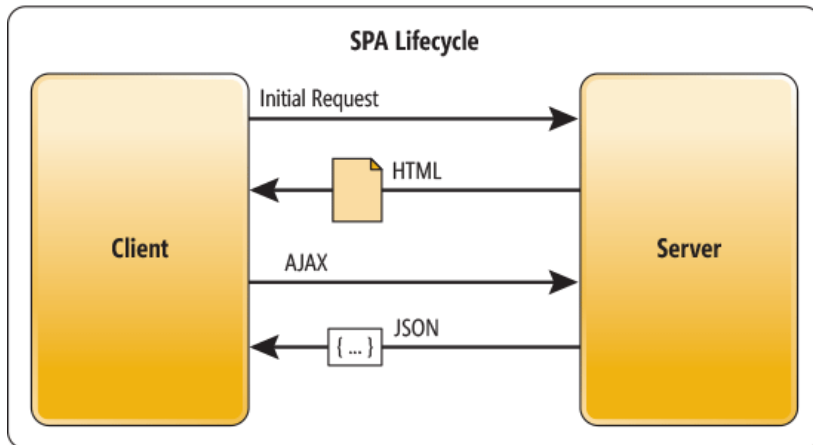




SPA vs MPA

SPA(Single Page Application)

- 한 개(Single)의 Page로 구성된 Application.
- 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.
- SPA는 보통 CSR(Client Side Rendering)방식으로 렌더링.

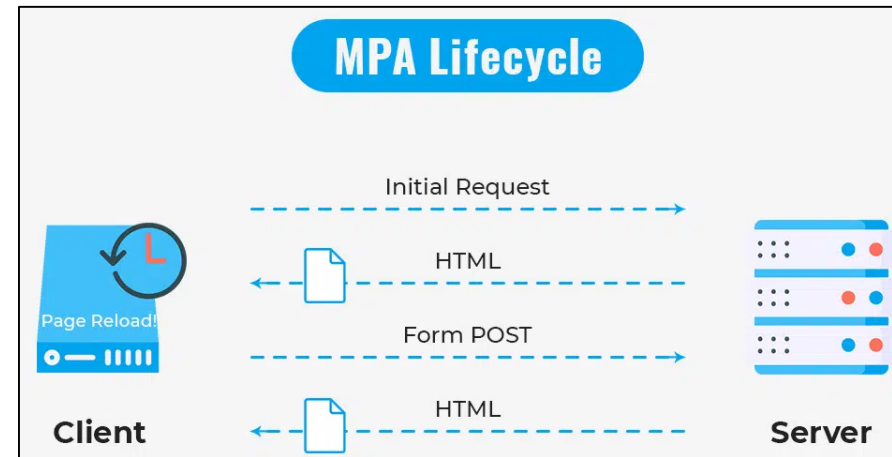
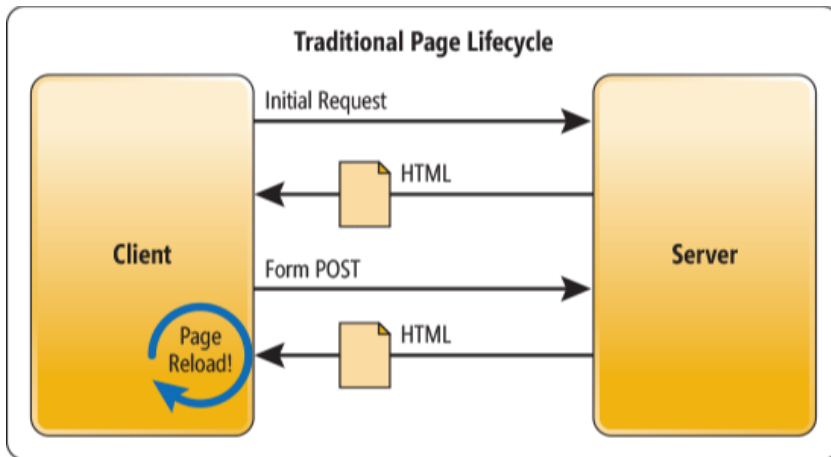




SPA vs MPA

MPA(Multi Page Application)

- 여러 개(Multiple)의 Page로 구성된 Application.
- 새로운 페이지를 요청할 때마다 서버에서 렌더링된 정적리소스가(HTML, CSS, JS)가 다운로드.
- 페이지를 이동하거나 새로고침하면 전체 페이지를 다시 렌더링.
- MPA는 SSR(Server Side Application)방식으로 렌더링.





SPA vs MPA

MPA(Multi Page Application) 장점

- MPA는 완성된 형태의 HTML 파일을 서버로부터 전달받는다.
- 서버에서 미리 렌더링을 하고 가져오기 때문에 첫 로딩이 매우 짧다.
- 그러나 클라이언트가 JS 파일을 모두 다운로드하고 적용하기 전까지는 각각의 기능은 동작하지 않는다.
- 다중 페이지로 원하는 만큼 페이지를 추가할 수 있다.
- 여러 페이지를 생성할 수 있기 때문에 훨씬 더 많은 수의 키워드를 타겟팅할 수 있다.
- 더 많은 페이지로 인해 여러 HTML 파일이 존재하는데, HTML 다운로드하여 페이지를 크롤링하기 때문에 검색 엔진이 작동하는 방식에 더 적합하다.
- 유저가 보는 화면 방식이 웹 크롤러가 보는 방식과 같아 SEO에 좀 더 적합하다.
- SPA보다는 오랜 역사를 가지고 있어서 해결법, 프레임워크 등 다양한 자료를 얻을 수 있다.



SPA vs MPA

MPA(Multi Page Application) 단점

- 새로운 페이지를 이동하면 '깜빡'인다. (UX)
- 새로운 페이지를 요청할 때마다 전체 페이지를 다시 렌더링하기 때문이다.
- 요청이 들어올 때마다 페이지를 새로 패치해야 하기 때문에 페이지 로딩 시간이 발생한다.
- 페이지 이동 시 불필요한 템플릿도 중복해서 로딩한다.
- 사용자가 새로운 페이지를 이동하면 전체 페이지를 다시 렌더링하기 때문이다.
- HTML, CSS, JS 와 같은 리소스들이 새로 고쳐져서 속도에 영향을 받는다.
- 클라이언트가 JS파일을 모두 다운로드하고 적용하기 전 까지 각 기능은 동작하지 않는다.
- 개발자는 클라이언트 측과 서버 측 모두에 프레임워크를 사용해야 합니다.
- 프론트와 백의 구분이 확연하지 않아 서버쪽, 클라이언트쪽 언어 모두 필요하다.
- 개발자는 모든 페이지가 안전한지 확인 해야하는데 페이지가 많아서 지속적으로 유지보수 하는 것이 힘들다.
- 모바일 앱 개발 시 추가적인 백엔드 작업 필요 (생산성)개발이 복잡해질 수 있다.
- 그 결과 애플리케이션 개발 시간이 더 길어진다.



SPA vs MPA

SPA(Single Page Application) 장점

- 전체 페이지를 업데이트 할 필요가 없기 때문에 빠르고 '깜빡' 거림이 없다.
- 반응형 디자인으로 페이지 상의 무언가를 클릭했을 시 기다림 없이 즉각적인 반응이 나타난다.
- 서버에게 정적 리소스를 최초 한번만 요청한다. 받은 데이터는 캐시로 전부 저장한다
- 모든 로컬 스토리지를 효과적으로 캐시할 수 있다. 애플리케이션은 하나의 요청만 보내고 모든 데이터를 저장한 다음 이 데이터를 사용할 수 있으며 오프라인에서도 작동한다.
- 모바일 앱 개발을 염두에 둔다면 동일한 API를 사용하도록 설계 가능하다.
- 전체 페이지를 다시 렌더링하지 않고 변경되는 부분만을 갱신하므로 새로고침이 발생하지 않아 네이티브 앱과 유사한 사용자 경험을 제공할 수 있다.
- 모바일 앱도 SPA와 동일한 아키텍처에서 개발되므로 모바일 개발을 염두해두면 동일한 백엔드 코드를 재사용하도록 할 수 있다.
- FE와 BE가 확실히 구분되어 백엔드 부분에 걱정할 필요가 없다.
- 서버의 템플릿 연산을 클라이언트로 분산이 가능하다.
- 컴포넌트 별 개발에 용이하고, 재사용이 가능하다.



SPA vs MPA

SPA(Single Page Application) 단점

- 최초 로딩 시 한번에 받아오기 때문에 초기 페이지 로딩시간이 길다(Webpack의 code splittion으로 해결가능).
- SPA는 JavaScript로 구축된다(CSR 방식). 자바스크립트를 읽지 못하는 검색엔진에 대해서 크롤링이 되지않아 색인이 되지 않는 문제가 발생할 수 있다.
- MPA보단 SEO에 대한 노출이 어렵다. 비동기로 렌더링되는 페이지이기 때문에 크롤러가 크롤링하기 어렵다(SSR, Next.js로 해결 가능).
- SSR에서는 서버에서 세션으로 관리하지만, CSR방식에서는 쿠키에 저장해야 하며 보안이 취약하다.
- XSS(교차 사이트 스크립팅)로 인해 공격자가 다른 사용자의 웹 응용 프로그램에 클라이언트 측 스크립트를 삽입할 수 있는 위험이 있다.



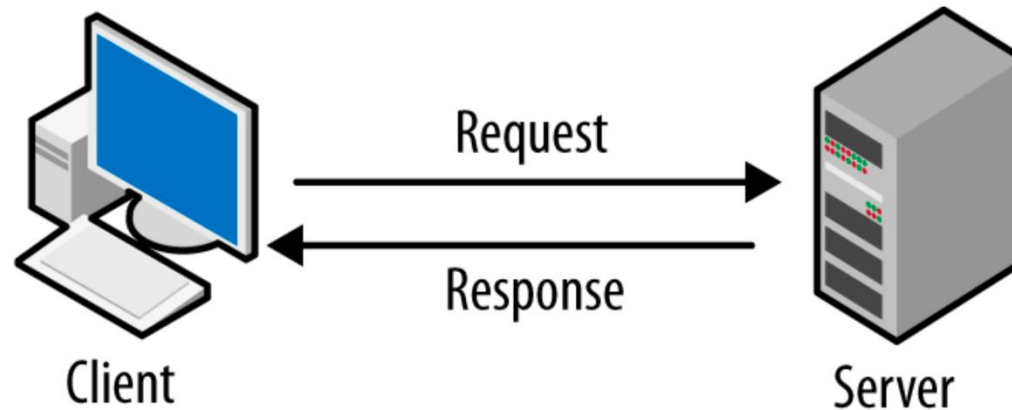
Client - Server

Client

- 특정 서비스를 이용하는 사용자(고객)
- 웹 브라우저

Server

- 특정 서비스를 제공하는 자





Browser

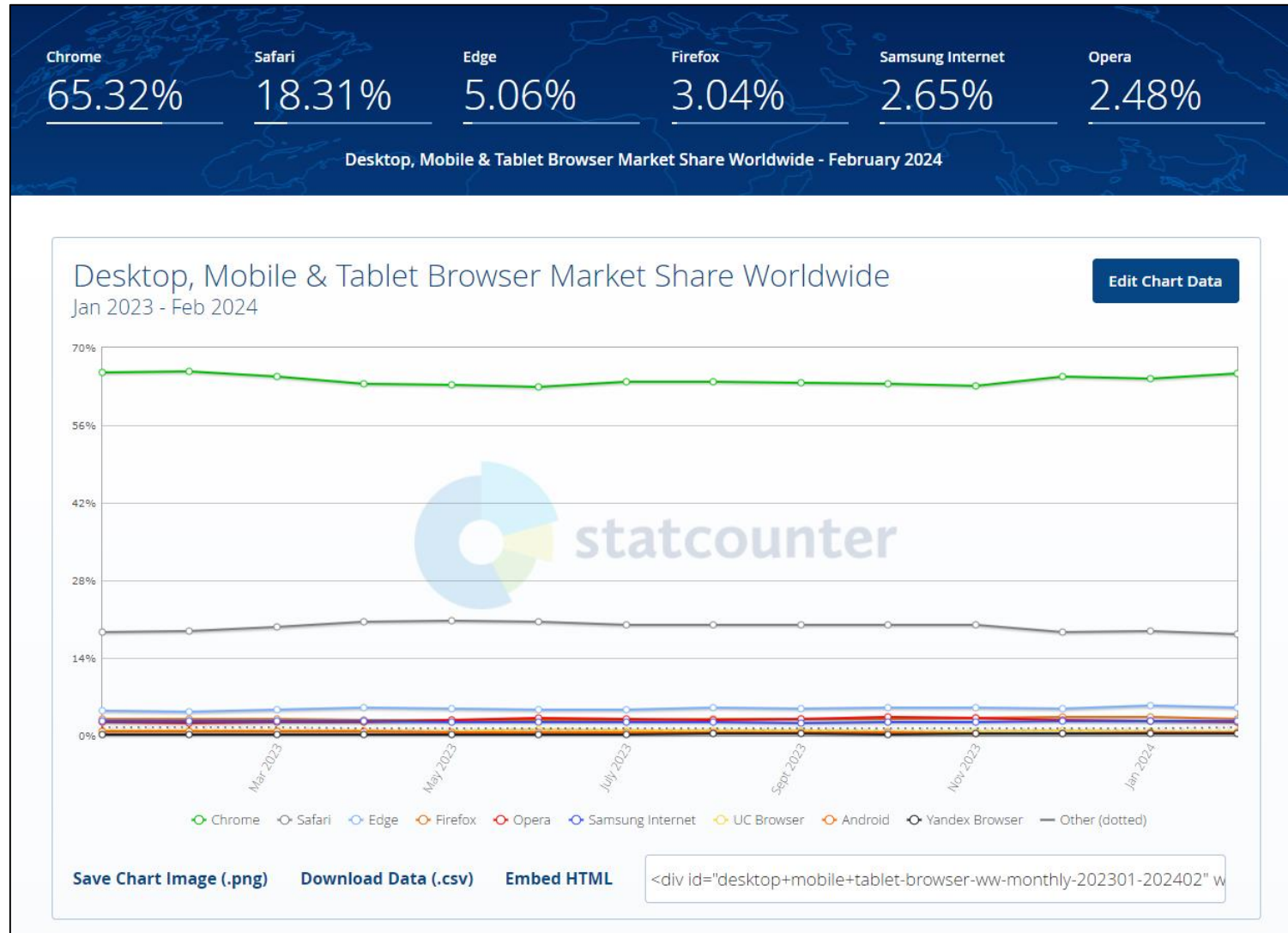
Web browser

- 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





Browser 점유율



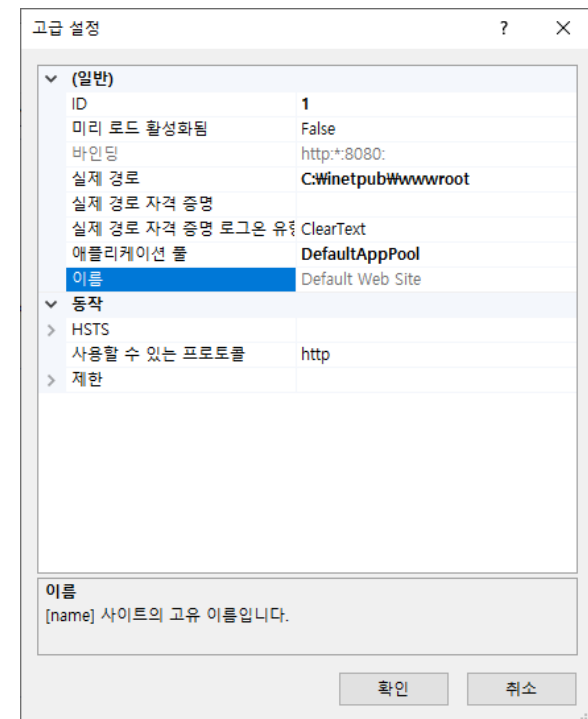


Web Server(IIS)



Window 10, Window11에는 IIS가 기본적으로 포함 되어 있음

- 시작 - "windows 기능 켜고 끄기" - 인터넷 정보 서비스 – World Wide Web 서비스
 - ✓ 보안 : 기본 인증, 요청 필터링 체크
 - ✓ 상태 및 진단 : HTTP 로깅, ODBC 로깅, 로깅 도구, 요청 모니터 체크
 - ✓ 성능 기능 : 모두 체크
 - ✓ 응용 프로그램 개발 기능 : 응용 프로그램 초기화를 제외한 모든 항목 체크
 - ✓ 일반적인 HTTP 기능 : 모두 체크
- 시작 – "IIS(인터넷 정보 서비스) 관리자"
 - Default Web Site – 고급 설정...
 - ✓ 실제 경로 : C:\inetpub\wwwroot
 - ✓ 바인딩 : 포트 번호 확인





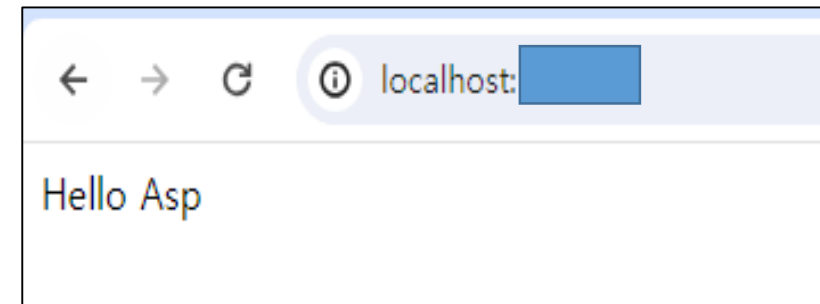
Web Server(IIS)

웹 서버 테스트

- C:\inetpub\wwwroot\default.asp 파일 작성
- 주소창(@d)에 "localhost:포트번호"

```
<%  
Response.Write("Hello Asp")  
%>
```

default.asp





Web Server(Tomcat)



Eclipse에서 Tomcat 연동하기

- Window – Preferences – Server – Runtime Environments – Add
 - Apache – Apache Tomcat v9.0 – Next – Download and Install(디렉토리 선택) – Finish – Apply and Close
- File – New – Dynamic Web Project
 - Project Name : "tomcatTest" – Finish
 - Package Explorer – tomcatTest – src – main – webapp – 마우스 우측 버튼 – New – JSP File – "index.jsp"

```
<%@ page language="java" contentType="text/html; charset=EUC-KR"
    pageEncoding="EUC-KR"%>
<!DOCTYPE html>
<html>
<body>
<%
    out.println("Helllo JSP");
%>
</body>
</html>
```

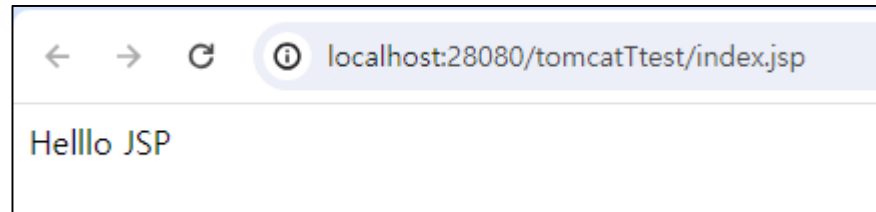
index.jsp



Web Server(Tomcat)

웹 서버 테스트

- Window – Show View – Servers – apache-tomcat-9.0.05 at localhost 더블클릭 – 포트 확인
(서버가 없는 경우 마우스 우측 버튼 – New – Apache – 서버 선택)
- Ctrl + F11 – 서버 선택 - Finish



Web Server(Apache)

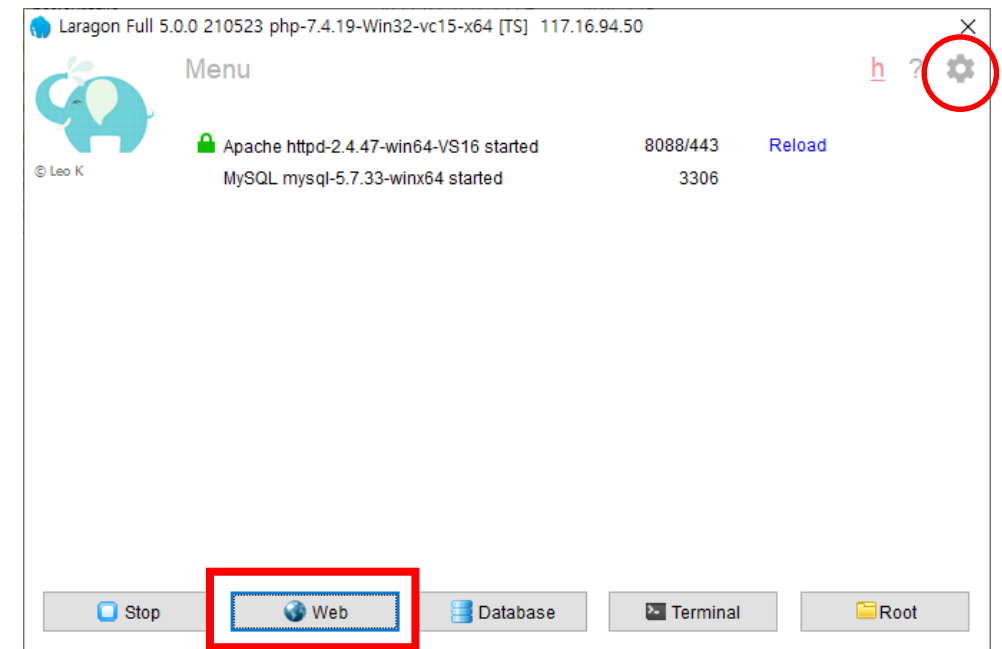
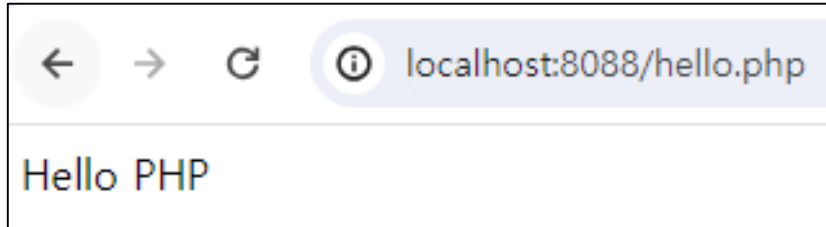


Laragon에서 Apache 사용하기

- C:\Wlaragon\www\ 폴더에 "hello.php"파일 생성
- 주소 창에 "localhost:포트번호/hello.php"

```
<?php
    echo "Hello PHP"
?>
```

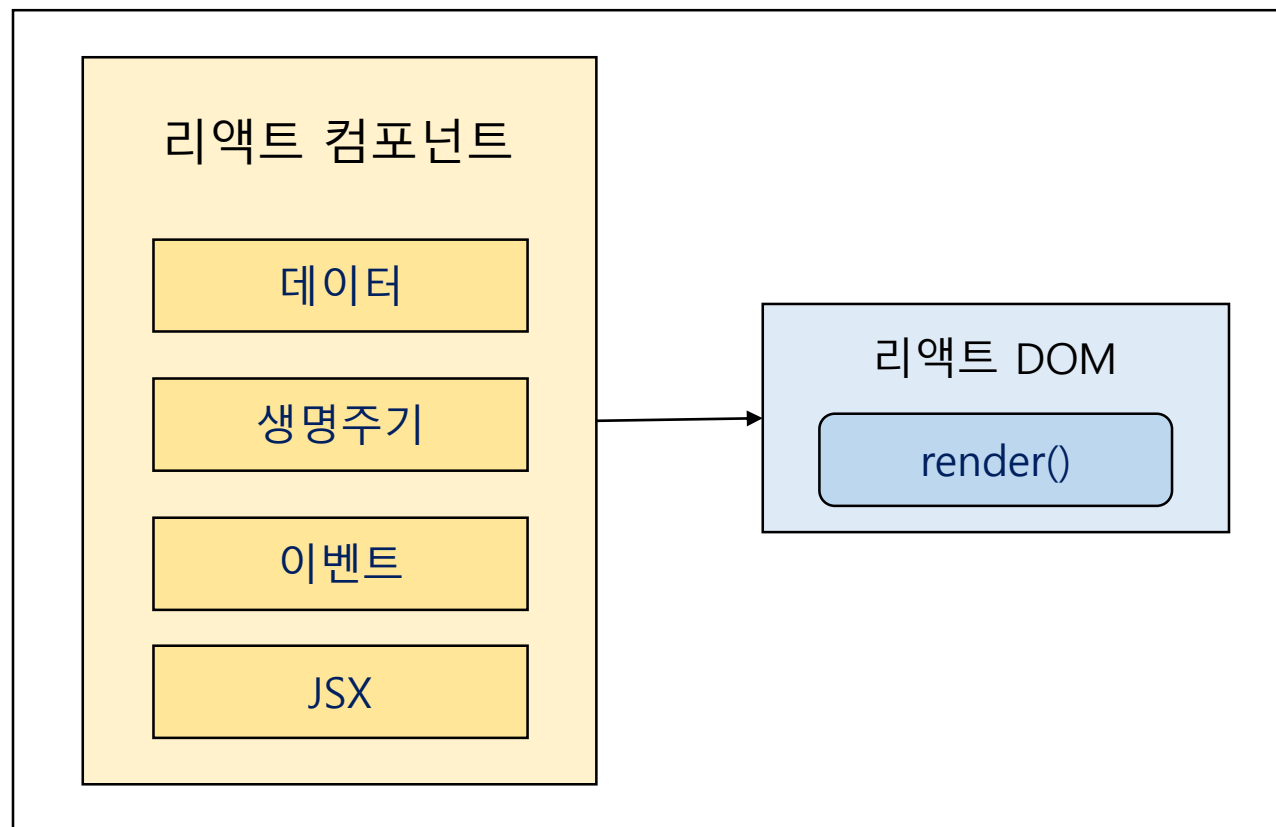
hello.php





React

The library for web and native user interfaces





Reference

- <https://www.inflearn.com/course/lecture?courseSlug=%EC%B2%98%EC%9D%8C-%EB%A7%8C%EB%82%9C-%EB%A6%AC%EC%95%A1%ED%8A%B8&unitId=112586>
- <https://developer.mozilla.org/ko/docs/Web/JavaScript>
- <https://javascript.info/>
- 명품 웹 프로그래밍, 황기태, 생능출판사
- 리액트 & 리액트 네이티브 통합 교과서, 아담 보두치, 강경일, 신희철, 에이콘