

# 풀스택 개발자를 위한 웹퍼블리싱

Webpublishing For Fullstack Developer

20211018  
Dosun, Kim

## STEP 6 구성 및 속성활용 2

# Hello, Web!

**월드 와이드 웹** (World Wide Web, WWW, W3)  
인터넷에 연결된 컴퓨터를 통해 사람들이 정보를 공유할 수 있는 **전 세계적인 정보 공간**을 말한다.

간단히 **웹**(the Web)이라 부르는 경우가 많다.

인터넷에서 HTTP 프로토콜, 하이퍼텍스트, 형식 등을 사용하여 그림과 문자를 **교환하는 전송 방식**을 말하기도 한다.

1

## 학습 로드맵 및 개요

웹트랜드 / 웹퍼블리싱 역할 / 웹표준 접근성 / 환경구성

2

## HTML5와 CSS3 이해

기본용어 정리 / 엘리먼트의 이해 및 활용 / 실습

3

## 레이아웃 구성과 속성활용 1

레이아웃별 컴포넌트 제작 ( 메뉴 / 콘텐츠 / 폼 / 테이블 )

4

## 레이아웃 구성과 속성활용 2

레이아웃별 컴포넌트 제작 ( 슬라이드 / 팝업 / 트리 / JS )

5

## 적응형 및 반응형 대응하기

<meta name="viewport"> / Media Queries 3-4

6

## 오픈소스 및 라이브러리 활용

Chart / Jqcloud / Bootstrap / Vue

# 라이브러리 or 프레임워크

## 집을 사거나, 혹은 조심스럽게 나만의 집을 짓거나

개발자들은 종종 "라이브러리"와 "프레임워크"라는 용어를 같은 것처럼 사용한다.

그러나 두 용어는 차이가 있다. 라이브러리와 프레임워크는 모두 일반적인 문제들을 더 쉽게 해결하기 위해 다른 개발자들이 작성한 재사용 가능한 코드이다. 이러한 개념의 차이점을 설명하기 위해 집을 예로 들어 설명할 수 있다.

**라이브러리**는 집을 직접 짓는 것과 같다. 당신이 원하는 구조를 선택하여 원하는 방식으로 방을 정렬할 수 있다. 반면 **프레임워크**는 새집을 사는 것과 같아서 구조를 직접 다룰 필요가 없지만, 집은 이미 지어져 있기 때문에 방을 정렬하는 방식을 선택할 수 없다.



React.js



ANGULAR



Vue.js

# 라이브러리 or 프레임워크

## 기술적 차이

프레임워크와 라이브러리의 기술적 차이점은 **제어 반전**(inversion of control)에 있다.

라이브러리를 사용할 때에는 어플리케이션 흐름을 직접 설정해야 한다.

라이브러리를 언제, 그리고 어디서 호출할 것인지는 사용자가 결정해야 한다.

그러나 프레임워크는 스스로 이러한 흐름을 담당한다.

프레임워크는 코드를 연결할 수 있는 위치를 제공하며, 필요에 따라 사용자가 연결한 코드를 호출한다.



React.js



Vue.js

# 라이브러리 or 프레임워크

## 프레임워크의 장점

일반적으로 프레임워크는 우리가 무엇을 해야 할지 알려준다.

프레임워크는 사용에 관한 **모범 사례**(best practice)를 가지고 있으며, 사용자를 지원하는 도구를 제공한다.

전담 팀에 의해 만들어진 이러한 프레임워크는 대규모 애플리케이션을 구축하는 데 필요한 모든 것들이 제공된다.

프레임워크의 개발팀은 모범 사례를 따르는 공식 스타일 가이드를 제공한다. 이러한 가이드를 통해 사용자는 빠르게 생산성을 높이고, 다른 팀원에 대한 교육도 쉽게 할 수 있다.



React.js



Vue.js

# 라이브러리 or 프레임워크

## 프레임워크의 단점

프레임워크는 많은 코드를 작성하여 구축된다. 즉, 로딩 시간이 길어지고 성능이 저하된다.

확장 가능한 아키텍처(Scalable architecture)는 위에서 수 많은 기능들을 제공한다.

일부 애플리케이션은 매우 간단하므로, 이러한 프레임워크의 사용이 자칫 개발 과정을 더욱 복잡하게 만들 수도 있다.

프레임워크를 사용하려면 지속해서 공부해야 한다. 새로운 버전이 나올 때마다 항목이 추가되고, 어떤 항목들은 제거되거나 더는 사용되지 않으므로 최신 지식을 유지해야 한다.



React.js



ANGULAR



Vue.js

# 라이브러리 or 프레임워크

## 라이브러리의 장점

라이브러리는 어떻게 사용할 것인가에만 집중되어 있다. 즉, HTTP, 라우팅, 등의 상태 관리를 위한 라이브러리는 지원하지 않는다.

이러한 특징 덕분에 사용자는 자신이 필요한 라이브러리만 직접 선택할 수 있다.

즉, 라이브러리를 사용하면 애플리케이션에 대한 통제권을 가질 수 있으며, 원하는 부분만 추가하면 되므로 제작과정이 더욱 원활 해진다.



React.js



Vue.js



# 라이브러리 or 프레임워크

## 라이브러리의 단점

특정 아키텍처를 직접 제작하다 보면 애플리케이션이 망가질 수 있으므로 주의해야 한다.

일부 사람들이 Angular 또는 Vue를 선택하는 이유는

자신만의 규칙을 구축하는 데 시간과 돈을 투자하고 싶지 않기 때문이다.

이러한 프레임워크를 사용하면 단순히 규칙을 배운 뒤, 실제 제작에만 집중할 수 있다.

전반적으로 프레임워크는 특정 방식에 따라야(opinionated) 하며, 라이브러리는 이에 비해 더욱 유연하다



React.js



ANGULAR



Vue.js

Thank you