

Angular vs React vs Vue

자바스크립트 라이브러리 전쟁

Angular JS

Angular

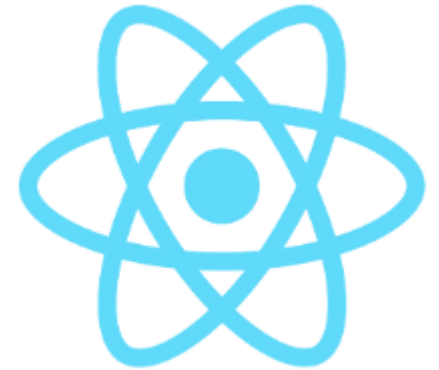
- 구글이라는 전세계적인 기업의 지원
- 가장 빠르게 릴리즈된 JS 프레임워크(2010년)
- 큰 커뮤니티
- 양방향 바인딩
- TypeScript 가장 먼저 도입
- 가장 체계적이고 잘 정리되어있는 문서
- 가장 복잡하고 큰 러닝커브 (내부 동작 메커니즘 등 포함)

HomePage : angularjs.org

Repository : github.com/angular/angular.js



React



React

- 웹 뿐만 아니라 앱 개발에도 사용 (React Native)
- 페이스북이라는 거대한 업체의 지원
- 가장 거대한 커뮤니티
- Virtual Dom

HomePage : reactjs.org/

Repository : github.com/facebook/react

Vue



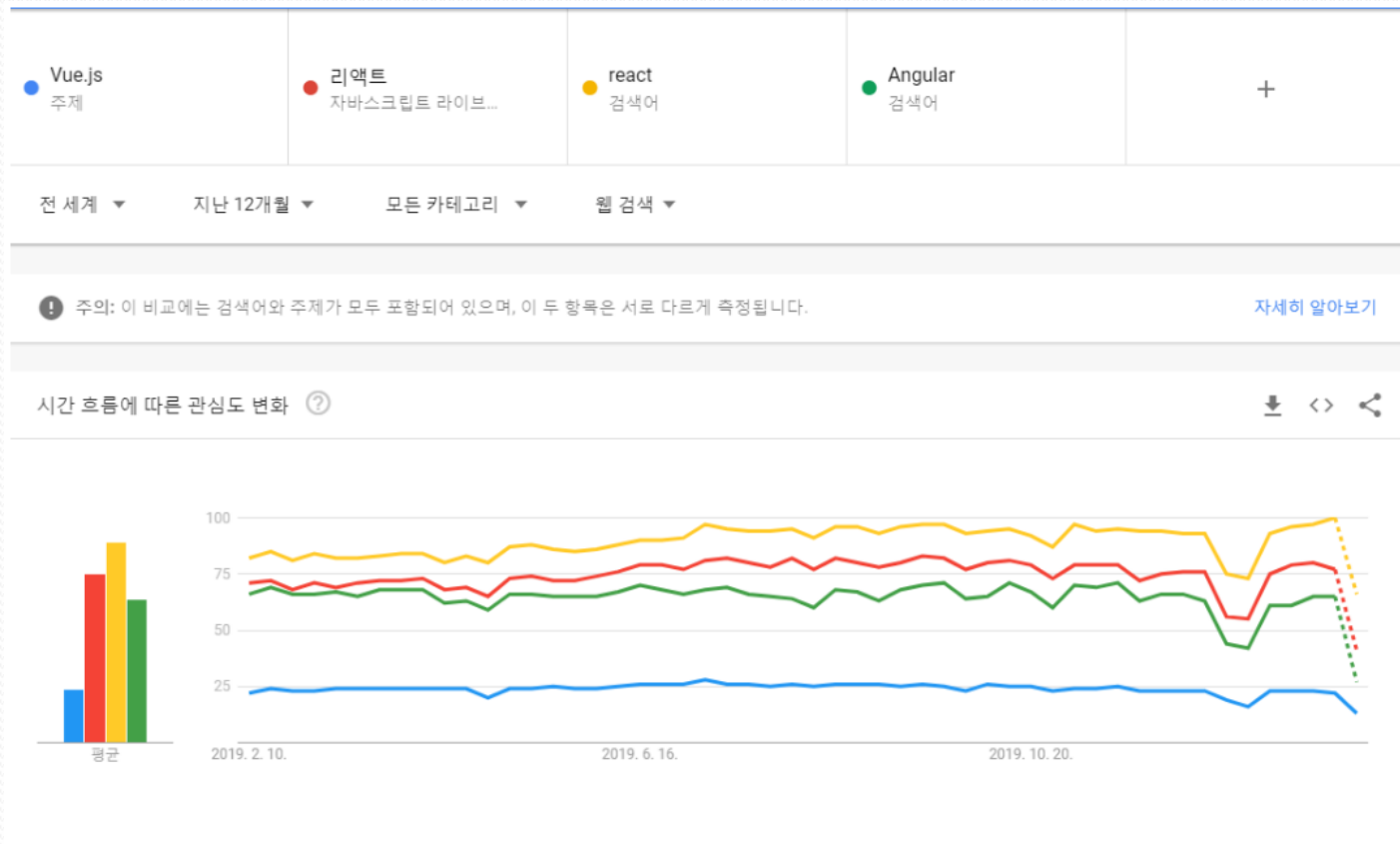
Vue

- 가장 적은 러닝커브 (jQuery 등에 익숙한 기존 개발자도 가장 편하게 도입 가능)
- 가볍고 빠르게 개발이 필요하다면 추천
- Evan You 라는 개인이 프로젝트 리더
- 2019년에 메인 리더인 Evan You 의 홍콩 시위 발언 관련하여 이슈 존재
- 상대적으로 작지만 빠르게 성장중인 커뮤니티
- 가장 최신에 릴리즈 시작한 프레임워크(2014년)
- 양방향 바인딩
- Virtual Dom

HomePage : github.com/vuejs/vue#readme

Repository : github.com/vuejs/vue

검색어 트렌드 비교



React vs Vue vs Angular For google trends

배우기 쉬운 정도

1. Learning curve

개발 경험에 따라, 개발자들은 어떤 프레임워크가 가장 배우기 쉬운지 의견이 다를 것이다.

- **난이도 하: Vue.js.** 내 생각에는, Vue.js가 대부분의 웹 개발자들이 쉽게 배울 수 있는 프레임워크이다. HTML과 Javascript의 기본에 가장 가깝기 때문에, 가장 친숙하다. Vue.js를 시작하는 것은, 당신의 HTML 문서에 하나를 추가하는 것만큼 쉽다. Vue.js는 당신이 더 복잡한 어플리케이션을 만들수록 더 복잡해진다. 복잡한 어플리케이션을 위해서는, 더 복잡한 프로젝트 설정(Vue CLI가 훨씬 더 쉽게 하지만) .vue 파일들을 다룰 줄 알아야한다. 그리고 더 복잡한 패턴을 파볼 필요가 있을 것이다.
- **난이도 상: Angular.js.** Angular는 Typescript를 사용하기 때문에, 배우기 어렵다. 컴포넌트, 모듈 그리고 구문들은 Javascript와는 꽤 다르다. Angular는 강력한 기능들이 많이 내장되어 있기 때문에, 개발자들이 어플리케이션을 개발하는데 꽤 도움이 될 수 있을지도 모르는 특정 패턴으로 코딩하도록 한다.
- **난이도 중: React.js.** "모든 것들이 Javascript이다"라는 접근이지만, 두 가지 주의할 점이 있다. 첫 번째로, ES6 구문들과 최적화되어 있기 때문에, 막 개발을 시작한 사람들에게는 어려울 수 있다. 두 번째는, Javascript의 대체적인 프로젝트에서 잘 쓰이지 않는, HTML과 Javascript이 혼합된 구문인 JSX이다. JSX는 HTML처럼 보이지만 Javascript이고, 코드를 이해하는데 혼란이 올 수 있기 때문에 처음에는 어려울 수 있다.

Learnig curve 승자는? Vue.js

단일 페이지 어플리케이션

2. Single-page apps (SPAs, 단일 페이지 앱)

Single-page Apps (SPAs)는 페이지 재로드 대기 시간이 거의 없을 정도로 반응적이고 모바일 웹 같다. 페이지는 브라우저에서 Javascript가 유저가 보는 모든 것들을 컨트롤하기 때문에 단일 HTML 파일이다. 버튼이 클릭되거나 새로운 윈도우가 페이지에 나타날 때, 서버로 어떠한 요청도 하지 않는다. 대신, Javascript 코드는 바로 수정하고 렌더링 한다. SPA는 매우 빠르게 로딩하기 때문에 인기가 높다.

Angular는 SPA를 만들기 위해 구글이 만들었다. TypeScript와 Angular의 내장된 툴들 그리고 서비스들은 개발자들에게 최고의 경험으로 SPA를 만들도록 해준다. 라우팅, 상태 관리 솔루션, 그리고 폼 유효성과 같은 도구들을 사용하면 "즉각적인" 개발 경험을 얻을 수 있다. 더 큰 SPA에 React.js 또는 Vue.js를 사용하면, Angular를 사용할 때보다 from 지원에 필요한 패키지의 이용과 같은 것들이 더 복잡할 수 있다.

Single-page app 승자는? Angular.js.

멀티 페이지 어플리케이션

3. Multi-page apps (MPAs, 멀티 페이지 앱)

Multi-page application (MPA)는 당신이 가장 좋아하는 e-commerce 또는 미디어 사이트 같은 "전통적인" 웹사이트이다. 사이트가 서버로 데이터를 보내거나 새로운 데이터를 표시할 때마다, 브라우저에 렌더링 될 요청을 서버에게 한다.

React.js는 페이지들의 기존 HTML을 향상하기 위해 쓰일 프로젝트들에는 잘 작동한다. JSX를 사용하여 React.js 구성 요소를 만드는 데는 주목할만한 가치가 있지만, 페이지에 강력한 Javascript 기반의 컴포넌트를 추가하기 위해 React.js를 사용할 수도 있다. 여전히 프론트엔드 빌드 워크플로우가 필요할 수도 있다. 그러나, Vue.js는 MPA를 만들기에 매우 간단하다. 이 기능들과 컴포넌트들을 페이지에 통합시키는 Vue.js 라이브러리를 импорт하면, 다른 과정은 필요하지 않다. 전반적으로, Vue.js, React.js 모두 MPA에 Javascript 기능들은 추가하기에 좋다.

내 생각에는 JSX와 같은 스텝들이 필요하지 않기 때문에, React.js보다는 Vue.js가 조금 더 낫다고 본다. (JSX는 옵션이지만 React.js 컴포넌트를 만드는 데에 표준이다.)

Angular.js로 MPA를 만드는 데는 추천하지 않는다. Angular의 컴포넌트들에 all-in-one 접근방식과 Typescript의 사용은 적합한 프레임워크가 아니다. HTML에 Angular 패키지들을 import 하는 것은 불필요하게 복잡할 것이다. HTML 기반의 앱인 MPA에 Typescript를 컴파일하는 것은 퍼포먼스에 영향을 주진 않겠지만, 개발 팀에게는 성가신 일이 될 것이다. 전반적으로, 미래에는 MPA가 더 실행 가능할만한 옵션이 될지도 모르지만, Angular는 SPA를 개발하기 위해 만들어졌다.

Multi-page app 승자는? React.js & Vue.js

퍼포먼스

4. Performance benchmarks

느린 로딩 타임과 저조한 퍼포먼스는 웹사이트에 중요한 비즈니스 리스크이다. 한 페이지를 모두 렌더링 하기 위해 유저를 몇 분 동안 기다리게 하는 것은 이미 경쟁자의 사이트로 옮길 찬스이다.

각 프레임워크들이 어떻게 작동하는지 조사하기 위해, 깃헙에서 [javascript 프레임워크 벤치마크](#)를 참고했다. 각 프레임워크가 꽤 빠르기 때문에, 이 벤치마크 결과는 모두 상대적이다. 한 시나리오에서 프레임 워크가 다른 프레임 워크보다 성능이 우수할 수 있지만, 벤치 마크에서 설명하지 않은 시나리오에서는 다른 성능을 볼 수 있기 때문에 벤치 마크 테스트는 다소 다루기 어렵다.

벤치마크 테스트 결과에 따르면(그림 1), Angular이 React.js와 Vue.js에 비해 시작 시 조금 더 느리다. 비록 Angular가 빠른 프레임워크 기는 하지만, 번들 사이즈가 크다. 그래서 큰 앱을 만들 때, React.js나 Vue.js보다 더 빠른 퍼포먼스를 보인다. React.js와 Vue.js는 작은 어플리케이션에서는 비슷하다.

Startup metrics

Name	vue-v2.5.16-keyed	angular-v6.0.0-keyed	react-v16.1.0-keyed
consistently interactive a pessimistic TTI - when the CPU and network are both definitely very idle. (no more CPU tasks over 50ms)	79.4 ± 1.0 (1.0)	137.1 ± 0.8 (1.7)	82.4 ± 1.0 (1.0)
script bootup time the total ms required to parse/compile/evaluate all the page's scripts	17.0 ± 0.5 (1.0)	55.9 ± 0.5 (3.3)	20.7 ± 0.4 (1.2)
main thread work cost total amount of time spent doing work on the main thread. Includes style/layout/etc.	177.8 ± 2.0 (1.0)	233.8 ± 2.0 (1.3)	181.9 ± 2.1 (1.0)
total byte weight network transfer cost (post-compression) of all the resources loaded into the page.	232,029.0 ± 0.0 (1.0)	384,701.0 ± 0.0 (1.7)	266,195.0 ± 0.0 (1.1)

Performance Benchmarks 승자는? 없음. 모두 훌륭한 런타임 퍼포먼스를 보인다.

배포하기 쉬운가?

5. Deployment

이 프레임 워크를 사용할 때 얼마나 쉽게 팀이 앱을 배포할 수 있을까?

배치하는데 전반적으로 가장 어려운 프레임워크는 **Angular**이지만, Angular CLI를 사용한다면, 이 프로세스는 대부분의 일이 되어있기 때문에 사실 간단하다. Angular CLI는 프로젝트 생성부터 코드 최적화까지 모든 것을 한다. 하나의 명령을 실행하면 모든 정적 호스트에 완전히 최적화된 번들 앱을 배포할 수 있다. 앱을 범용(서버 측 사전 렌더링 된 앱)으로 전환하는 것 또한 한 번의 명령으로 가능하다.

React.js 앱을 빌드하고 배포하는 것 또한 쉽다. **create-react-app**을 사용하면, 개발 및 최종 프로젝트 구축에 필요한 모든 것을 포함하는 완전히 사전 구성된 새로운 React.js 앱을 생성하는 도구가 있습니다. 주어진 구성으로 모든 정적 호스트에 배포할 수 있는 완전히 최적화된 프로젝트를 만들 수 있다. Angular CLI와는 달리 명령어 하나로 서버사이드 렌더링은 할 수 없다.

Vue.js의 경우, CLI로 사전 구성된 모든 것들을 포함한 강력한 프로젝트를 구성할 수 있다. 개발이 쉽고 Angular, React.js처럼 내장된 명령어를 실행함으로써 최적화된 프로젝트를 시작할 수 있다. Vue.js 프로젝트는 어떤 정적인 호스트에도 배포될 수 있고, 서버사이드 렌더링(SEO; Search Engine Optimization)을 켜는 것이 비교적 쉽다.

Deployment 승자는? 없음. 모든 프레임워크들이 관리하고 배포하기에 쉬움.

기업 인기도

6. Industry adoption

Stackoverflow의 2019 개발자 설문에 따르면, React.js와 Vue.js가 각각 첫 번째, 두 번째로 가장 사랑받은 웹 프레임워크로 선정되었다. Vue.js는 업계에서 두 번째로 개발자들이 가장 많이 쓰이는 웹 프레임워크로 뽑히면서 그 수요가 강조됐다.

2019년 8월 기준으로, Indeed.com은 미국의 15,000여 직업들이 React.js 스킬을 필요한 것으로 보여줬다. 이어, 13,500여 개의 Angular, 그리고 2,200여 개의 Vue.js 포지션이 있다. React.js는 아마도 엔지니어링 팀 리더들과 개발자들이 웹 개발 스킬을 늘리기 위한 가장 적합한 프레임워크인 듯하다.

Industry adoption 승자는? React.js & Angular