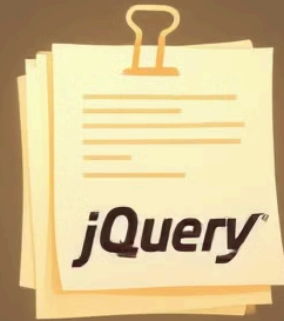


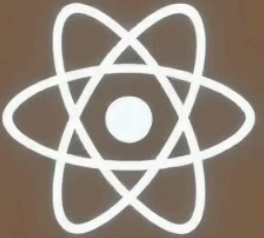
React 개발배경

jQuery에서 React로의 전환 배경과 React의 핵심 개념을 살펴보겠습니다.

 by **Hank Kim**



0
1
1+
+ 2_{x=}
react.0 +
2
0
0



jQuery에서 React로의 전환 배경



명령형 프로그래밍의 한계

jQuery는 "어떻게" 화면을 변경할지 일일이 지시해야 했습니다. 복잡한 UI 상태 관리가 어려웠습니다.



코드 재사용성 부족

동일한 기능을 여러 곳에서 사용하려면 코드를 중복해야 했습니다.



유지보수 어려움

애플리케이션 규모가 커질수록 코드 관리가 어려워졌습니다.



성능 병목 현상

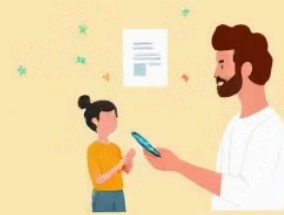
대규모 애플리케이션에서 잦은 DOM 조작은 성능 저하를 가져왔습니다.

React is Evolution

Initial Development
2011.: Initial Development



2015: React Native
2015: Open Source Release



2015: React Native



PRESENT
WIDESPREAD ADOPTION
Cartiuaile with Elegtraas Adoption



PRESENT: CONFEPREAD ADOPTION
THIS SUCCESS OF REACT



React의 탄생

1

2011년

페이스북 내부에서 Jordan Walke에 의해 개발 시작

2

2013년

JSConf에서 React 첫 공개

3

초기 반응

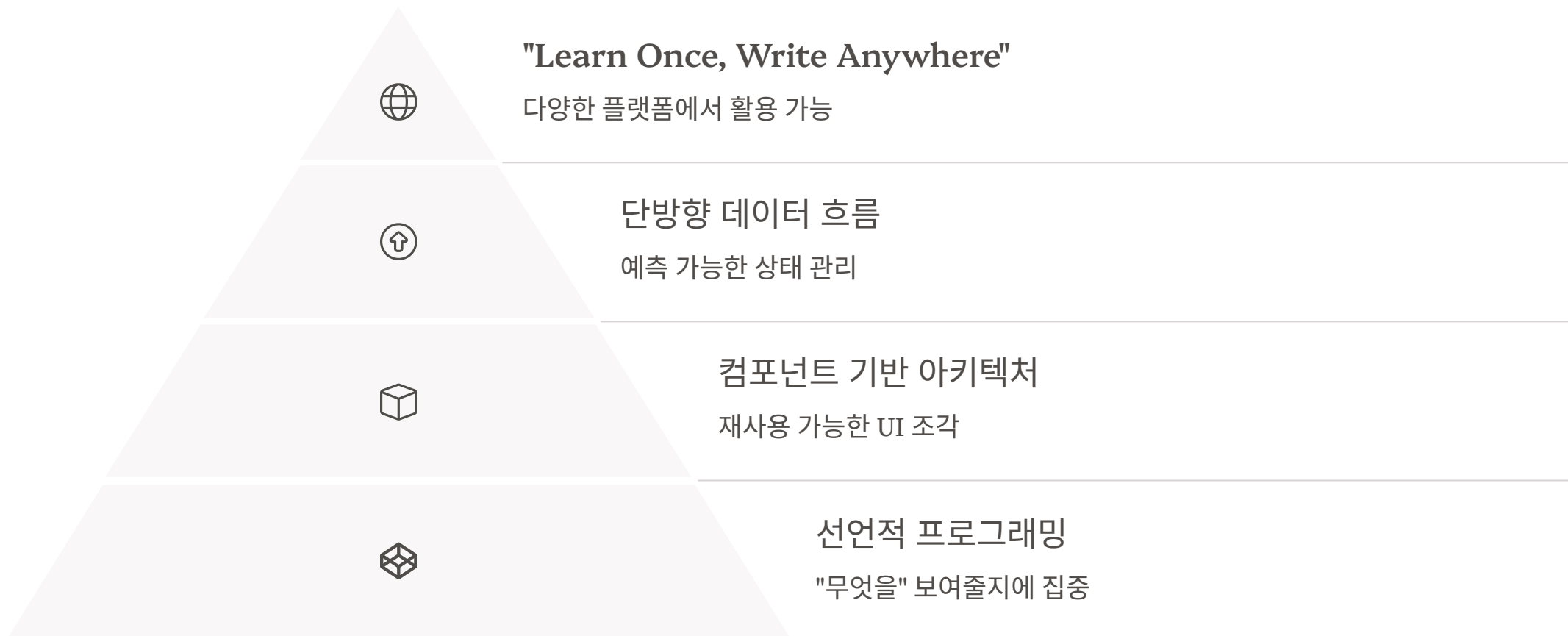
JSX 문법에 대한 의구심

4

검증

페이스북, 인스타그램에서 가치 입증

React의 개발 철학



React의 주요 기능

JSX

JavaScript 내에서 XML 형태의 마크업을 작성할 수 있는 문법 확장입니다. 직관적인 코드 구조를 제공합니다.

컴포넌트

UI의 독립적인 조각으로, 자체 로직과 렌더링 방식을 가집니다. 함수형과 클래스형이 있습니다.

Props와 State

Props는 부모에서 자식으로 데이터를 전달합니다. State는 컴포넌트 내부에서 관리되는 데이터입니다.

Hooks

함수형 컴포넌트에서 상태와 생명주기 기능을 사용할 수 있게 해주는 API입니다.

Virtual DOM 개념



가상 DOM 생성

데이터가 변경되면 전체 UI를 Virtual DOM에 렌더링합니다.



비교 (Diffing)

이전 Virtual DOM과 새로운 Virtual DOM을 비교합니다.



변경 사항 식별

실제로 변경된 부분만 식별합니다.



실제 DOM 업데이트

변경된 부분만 실제 DOM에 적용합니다.

Diffing 알고리즘

트리 비교

서로 다른 타입의 두 요소는 다른 트리를 생성합니다.



리스트 렌더링

고유하고 안정적인 key 설정이 중요합니다.



Key 속성

개발자는 key prop으로 안정적인 항목을 식별할 수 있습니다.



성능 최적화

$O(n^3)$ 에서 $O(n)$ 으로 시간 복잡도를 개선했습니다.



Reconciliation (재조정)

렌더링 단계 (Render Phase)

컴포넌트를 호출하고 변경 사항을 계산합니다.
비동기적으로 수행될 수 있습니다.

커밋 단계 (Commit Phase)

계산된 변경 사항을 실제 DOM에 적용합니다.
항상 동기적으로 수행됩니다.

Fiber 아키텍처

렌더링 작업을 작은 단위로 나누어 우선순위를 지정합니다. 작업을 중단하고 재개할 수 있습니다.