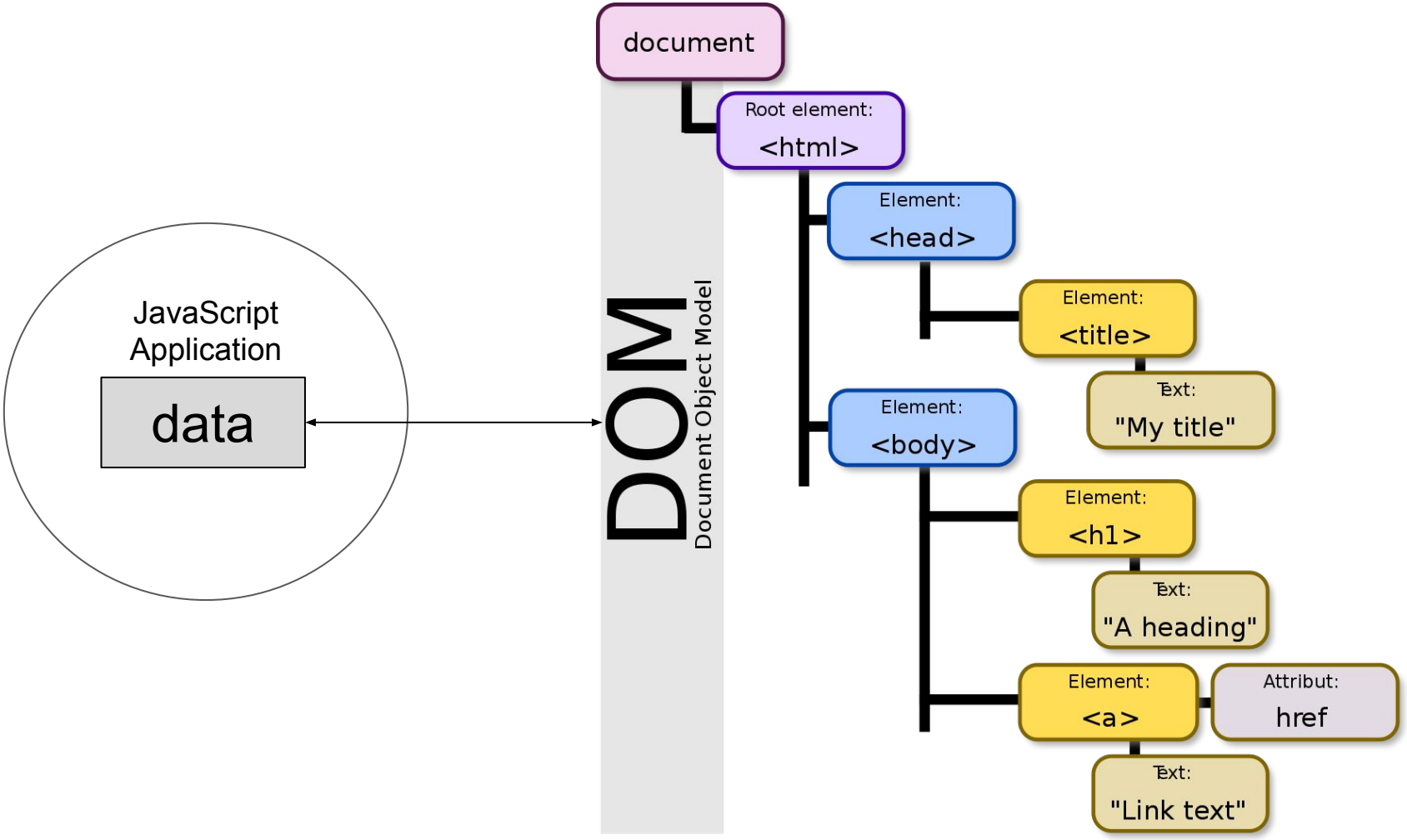
An Introduction to React


JavaScript
Application
data
DOM
Document Object Model
document
Root element:
<html>
Element:
<head>
Element:
<title>
Text:
"My title"
Element:
<body>
Element:
<h1>
Text:
"A heading"
Element:
<a>
Attribut:
href
Text:
"Link text"

React: a JS UI library

- Composable
- Declarative
- Efficient
- Flexible
- Doesn’t have a complex two-way data flow
- Widely used in the industry
- React Native

Composable

- Compose complex UIs from small and isolated pieces of code called components
- Build encapsulated components that manage their own state
- Component logic is written in JavaScript instead of templates so you can keep state out of the DOM.

Declarative

- App = f : data → DOM
- Tell it what you want to see
- Just design views for each state in your application, React will take care of the rest
- Declarative views make your code more predictable and easier to debug

Efficient

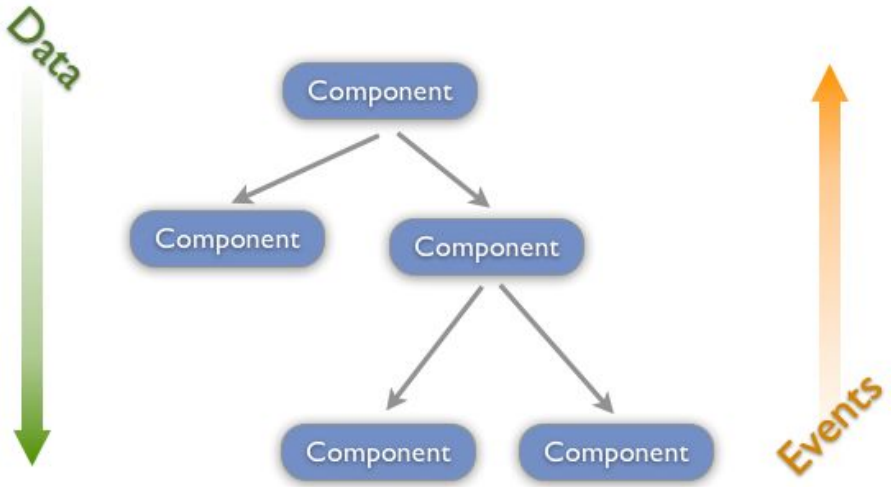
- React has an in-memory representation of the browser's DOM called the virtual DOM.
- When changes to the application state are made, React compares the virtual DOM with the browser DOM.
- Pushes the smallest needed changes to the DOM.

Flexible

- React is a library (not a framework)
- No opinions on:
  - Server communication
  - Routing
  - Data management
- Big and active ecosystem
- Starter projects

No complex two-way data flow

- Uses simple one-way reactive data flow
- Easier to understand than two-way binding
- Uses less code



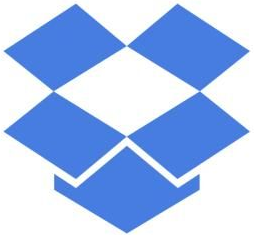
facebook

UniProt

airbnb

YAHOO!

Khan Academy



React Native

- Develop mobile phone applications
- Background thread interprets JS over a bridge
- DOM replaced by native views

Live coding

- Hello, world!
- Message
- Counter
- Counters

Concepts demonstrated

- Component
- JSX
- Props
- Styling
- State
- Render lifecycle
- Functional component