WIKIPEDIA

React (JavaScript library)

React (also known as **React.js** or **ReactJS**) is a free and open-source front-end JavaScript library[3] for building user interfaces based on UI components. It is maintained by Meta (formerly Facebook) and a community of individual developers and companies.[4][5][6] React can be used as a base in the development of single-page or mobile applications. However, React is only concerned with state management and rendering that state to the DOM, so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality.[7]

React

Original author(s)	Jordan Walke
Developer(s)	Meta and community
Initial release	May 29, 2013[1]
Stable release	17.0.2[2] / 22 March 2021
Repository	github.com/facebook/react (https://github.com/facebook/react)
Written in	JavaScript
Platform	Web platform
Type	JavaScript library
License	MIT License
Website	reactjs.org (https://reactjs.org/)

Contents

Basic usage

The following is a rudimentary example of React usage in HTML with JSX and JavaScript.

```
<div id="myReactApp"></div>

<script type="text/babel">
  function Greeter(props) {
```

```
    return <h1>{props.greeting}</h1>;
  }
  let App = <Greeter greeting="Hello, World!" />;
  ReactDOM.render(App, document.getElementById('myReactApp'));
</script>
```

The `Greeter` function is a React component that accepts a property `greeting`. The variable `App` is an instance of the `Greeter` component where the `greeting` property is set to `'Hello, World!'`. The `ReactDOM.render` method then renders the Greeter component inside the DOM element with id `myReactApp`.

When displayed in a web browser the result will be:

```
<div id="myReactApp">
  <h1>Hello, World!</h1>
</div>
```

Notable features

Declarative

React adheres to the declarative programming paradigm. Developers design views for each state of an application, and React updates and renders components when data changes.

Components

React code is made of entities called components. Components can be rendered to a particular element in the DOM using the React DOM library. When rendering a component, one can pass in values that are known as "props":[8]

```
ReactDOM.render(<Greeter greeting="Hello World!" />, document.getElementById('myReactApp'));
```

The two primary ways of declaring components in React is via function components and class-based components.

Function components

Function components are declared with a function that then returns some JSX.

```
const Greeter = (props) => <div>Hello, {props.name}!</div>;
```

Class-based components

Class-based components are declared using ES6 classes.

```
class ParentComponent extends React.Component {
  state = { color: 'green' };
  render() {
    return (
      <ChildComponent color={this.state.color} />
    );
```

```
  }
}
```

Virtual DOM

Another notable feature is the use of a virtual Document Object Model, or virtual DOM. React creates an in-memory data-structure cache, computes the resulting differences, and then updates the browser's displayed DOM efficiently.[9] This process is called **reconciliation**. This allows the programmer to write code as if the entire page is rendered on each change, while the React libraries only render subcomponents that actually change. This selective rendering provides a major performance boost.[10] It saves the effort of recalculating the CSS style, layout for the page and rendering for the entire page.[10]

Lifecycle methods

Lifecycle methods use a form of hooking that allows the execution of code at set points during a component's lifetime.

- `shouldComponentUpdate` allows the developer to prevent unnecessary re-rendering of a component by returning false if a render is not required.
- `componentDidMount` is called once the component has "mounted" (the component has been created in the user interface, often by associating it with a DOM node). This is commonly used to trigger data loading from a remote source via an API.
- `componentWillUnmount` is called immediately before the component is torn down or "unmounted". This is commonly used to clear resource-demanding dependencies to the component that will not simply be removed with the unmounting of the component (e.g., removing any `setInterval()` instances that are related to the component, or an "eventListener" set on the "document" because of the presence of the component)
- `render` is the most important lifecycle method and the only required one in any component. It is usually called every time the component's state is updated, which should be reflected in the user interface.

JSX

JSX, or JavaScript XML, is an extension to the JavaScript language syntax.[11] Similar in appearance to HTML, JSX provides a way to structure component rendering using syntax familiar to many developers. React components are typically written using JSX, although they do not have to be (components may also be written in pure JavaScript). JSX is similar to another extension syntax created by Facebook for PHP called XHP.

An example of JSX code:

```
class App extends React.Component {
  render() {
    return (
      <div>
        <p>Header</p>
        <p>Content</p>
        <p>Footer</p>
      </div>
    );
  }
}
```

Architecture beyond HTML

The basic architecture of React applies beyond rendering HTML in the browser. For example, Facebook has dynamic charts that render to `<canvas>` tags,[12] and Netflix and PayPal use universal loading to render identical HTML on both the server and client.[13][14]

React hooks

Hooks are functions that let developers "hook into" React state and lifecycle features from function components.[15] Hooks do not work inside classes — they let you use React without classes.[16]

React provides a few built-in hooks like `useState`,[17] `useContext`, `useReducer` , `useMemo` and `useEffect`.[18] Others are documented in the Hooks API Reference.[19] `useState` , `useReducer` and `useEffect`, which are the most used, are for controlling state and side effects respectively.

Rules of hooks

There are rules of hooks[20] which describe the characteristic code pattern that hooks rely on. It is the modern way to handle state with React.

1. Hooks should only be called at the top level (not inside loops or if statements).
2. Hooks should only be called from React function components, not normal functions or class components.

Although these rules can't be enforced at runtime, code analysis tools such as linters can be configured to detect many mistakes during development. The rules apply to both usage of hooks and the implementation of custom hooks,[21] which may call other hooks.

Common idioms

React does not attempt to provide a complete "application library". It is designed specifically for building user interfaces[3] and therefore does not include many of the tools some developers might consider necessary to build an application. This allows the choice of whichever libraries the developer prefers to accomplish tasks such as performing network access or local data storage. Common patterns of usage have emerged as the library matures.

Unidirectional data flow

To support React's concept of unidirectional data flow (which might be contrasted with AngularJS's bidirectional flow), the Flux architecture was developed as an alternative to the popular model–view–controller architecture. Flux features *actions* which are sent through a central *dispatcher* to a *store*, and changes to the store are propagated back to the view.[22] When used with React, this propagation is accomplished through component properties. Since its conception, Flux has been superseded by libraries such as Redux and MobX.[23]

Flux can be considered a variant of the observer pattern.[24]

A React component under the Flux architecture should not directly modify any props passed to it, but should be passed callback functions that create *actions* which are sent by the dispatcher to modify the store. The action is an object whose responsibility is to describe what has taken place:

for example, an action describing one user "following" another might contain a user id, a target user id, and the type `USER_FOLLOWED_ANOTHER_USER`.[25] The stores, which can be thought of as models, can alter themselves in response to actions received from the dispatcher.

This pattern is sometimes expressed as "properties flow down, actions flow up". Many implementations of Flux have been created since its inception, perhaps the most well-known being Redux, which features a single store, often called a single source of truth.[26]

Future development

Project status can be tracked via the core team discussion forum.[27] However, major changes to React go through the Future of React repository issues and pull requests.[28][29] This enables the React community to provide feedback on new potential features, experimental APIs and JavaScript syntax improvements.

History

React was created by Jordan Walke, a software engineer at Facebook, who released an early prototype of React called "FaxJS".[30][31] He was influenced by XHP, an HTML component library for PHP. It was first deployed on Facebook's News Feed in 2011 and later on Instagram in 2012.[32] It was open-sourced at JSConf US in May 2013.[31]

React Native, which enables native Android, iOS, and UWP development with React, was announced at Facebook's React Conf in February 2015 and open-sourced in March 2015.

On April 18, 2017, Facebook announced React Fiber, a new set of internal algorithms for rendering, as opposed to React's old rendering algorithm, Stack.[33] React Fiber was to become the foundation of any future improvements and feature development of the React library.[34] The actual syntax for programming with React does not change; only the way that the syntax is executed has changed.[35] React's old rendering system, Stack, was developed at a time when the focus of the system on dynamic change was not understood. Stack was slow to draw complex animation, for example, trying to accomplish all of it in one chunk. Fiber breaks down animation into segments that can be spread out over multiple frames. Likewise, the structure of a page can be broken into segments that may be maintained and updated separately. JavaScript functions and virtual DOM objects are called "fibers", and each can be operated and updated separately, allowing for smoother on-screen rendering.[36]

On September 26, 2017, React 16.0 was released to the public.[37]

On February 16, 2019, React 16.8 was released to the public.[38] The release introduced React Hooks.[39]

On August 10, 2020, the React team announced the first release candidate for React v17.0, notable as the first major release without major changes to the React developer-facing API.[40]

Versions

Version	Release Date	Changes
0.3.0	29 May 2013	Initial Public Release
0.4.0	20 July 2013	Support for comment nodes `<div>{/* */}</div>`, Improved server-side rendering APIs, Removed React.autoBind, Support for the key prop, Improvements to forms, Fixed bugs.
0.5.0	20 October 2013	Improve Memory usage, Support for Selection and Composition events, Support for getInitialState and getDefaultProps in mixins, Added React.version and React.isValidClass, Improved compatibility for Windows.
0.8.0	20 December 2013	Added support for rows & cols, defer & async, loop for `<audio>` & `<video>`, autoCorrect attributes. Added onContextMenu events, Upgraded jstransform and esprima-fb tools, Upgraded browserify.
0.9.0	20 February 2014	Added support for crossOrigin, download and hrefLang, mediaGroup and muted, sandbox, seamless, and srcDoc, scope attributes, Added any, arrayOf, component, oneOfType, renderable, shape to React.PropTypes, Added support for onMouseOver and onMouseOut event, Added support for onLoad and onError on `<img>` elements.
0.10.0	21 March 2014	Added support for srcSet and textAnchor attributes, add update function for immutable data, Ensure all void elements don't insert a closing tag.
0.11.0	17 July 2014	Improved SVG support, Normalized e.view event, Update $apply command, Added support for namespaces, Added new transformWithDetails API, includes pre-built packages under dist/, MyComponent() now returns a descriptor, not an instance.
0.12.0	21 November 2014	Added new features Spread operator ({...}) introduced to deprecate this.transferPropsTo, Added support for acceptCharset, classID, manifest HTML attributes, React.addons.batchedUpdates added to API, @jsx React.DOM no longer required, Fixed issues with CSS Transitions.
0.13.0	10 March 2015	Deprecated patterns that warned in 0.12 no longer work, ref resolution order has changed, Removed properties this._pendingState and this._rootNodeID, Support ES6 classes, Added API React.findDOMNode(component), Support for iterators and immutable-js sequences, Added new features React.addons.createFragment, deprecated React.addons.classSet.
0.14.1	29 October 2015	Added support for srcLang, default, kind attributes, and color attribute, Ensured legacy .props access on DOM nodes, Fixed scryRenderedDOMComponentsWithClass, Added react-dom.js.
15.0.0	7 April 2016	Initial render now uses document.createElement instead of generating HTML, No more extra `<span>`s, Improved SVG support, `ReactPerf.getLastMeasurements()` is opaque, New deprecations introduced with a warning, Fixed multiple small memory leaks, React DOM now supports the cite and profile HTML attributes and cssFloat, gridRow and gridColumn CSS properties.
15.1.0	20 May 2016	Fix a batching bug, Ensure use of the latest object-assign, Fix regression, Remove use of merge utility, Renamed some modules.
15.2.0	1 July 2016	Include component stack information, Stop validating props at mount time, Add React.PropTypes.symbol, Add onLoad handling to `<link>` and onError handling to `<source>` element, Add `isRunning()` API, Fix performance regression.
15.3.0	30 July 2016	Add React.PureComponent, Fix issue with nested server rendering, Add xmlns, xmlnsXlink to support SVG attributes and referrerPolicy to HTML attributes, updates React Perf Add-on, Fixed issue with ref.
15.3.1	19 August 2016	Improve performance of development builds, Cleanup internal hooks, Upgrade fbjs, Improve startup time of React, Fix memory leak in server rendering, fix React Test Renderer, Change trackedTouchCount invariant into a console.error.
15.4.0	16 November 2016	React package and browser build no longer includes React DOM, Improved development performance, Fixed occasional test failures, update batchedUpdates API, React Perf, and `ReactTestRenderer.create()`.
15.4.1	23 November	Restructure variable assignment, Fixed event handling, Fixed compatibility of browser build with AMD environments.

	2016	
15.4.2	6 January 2017	Fixed build issues, Added missing package dependencies, Improved error messages.
15.5.0	7 April 2017	Added react-dom/test-utils, Removed peerDependencies, Fixed issue with Closure Compiler, Added a deprecation warning for React.createClass and React.PropTypes, Fixed Chrome bug.
15.5.4	11 April 2017	Fix compatibility with Enzyme by exposing batchedUpdates on shallow renderer, Update version of prop-types, Fix react-addons-create-fragment package to include loose-envify transform.
15.6.0	13 June 2017	Add support for CSS variables in style attribute and Grid style properties, Fix AMD support for addons depending on react, Remove unnecessary dependency, Add a deprecation warning for React.createClass and React.DOM factory helpers.
16.0.0	26 September 2017	Improved error handling with introduction of "error boundaries", React DOM allows passing non-standard attributes, Minor changes to setState behavior, remove react-with-addons.js build, Add React.createClass as create-react-class, React.PropTypes as prop-types, React.DOM as react-dom-factories, changes to the behavior of scheduling and lifecycle methods.
16.1.0	9 November 2017	Discontinuing Bower Releases, Fix an accidental extra global variable in the UMD builds, Fix onMouseEnter and onMouseLeave firing, Fix <textarea> placeholder, Remove unused code, Add a missing package.json dependency, Add support for React DevTools.
16.3.0	29 March 2018	Add a new officially supported context API, Add new packagePrevent an infinite loop when attempting to render portals with SSR, Fix an issue with this.state, Fix an IE/Edge issue.
16.3.1	3 April 2018	Prefix private API, Fix performance regression and error handling bugs in development mode, Add peer dependency, Fix a false positive warning in IE11 when using Fragment.
16.3.2	16 April 2018	Fix an IE crash, Fix labels in User Timing measurements, Add a UMD build, Improve performance of unstable_observedBits API with nesting.
16.4.0	24 May 2018	Add support for Pointer Events specification, Add the ability to specify propTypes, Fix reading context, Fix the `getDerivedStateFromProps()` support, Fix a testInstance.parent crash, Add React.unstable_Profiler component for measuring performance, Change internal event names.
16.5.0	5 September 2018	Add support for React DevTools Profiler, Handle errors in more edge cases gracefully, Add react-dom/profiling, Add onAuxClick event for browsers, Add movementX and movementY fields to mouse events, Add tangentialPressure and twist fields to pointer event.
16.6.0	23 October 2018	Add support for contextType, Support priority levels, continuations, and wrapped callbacks, Improve the fallback mechanism, Fix gray overlay on iOS Safari, Add `React.lazy()` for code splitting components.
16.7.0	20 December 2018	Fix performance of React.lazy for lazily-loaded components, Clear fields on unmount to avoid memory leaks, Fix bug with SSR, Fix a performance regression.
16.8.0	6 February 2019	Add Hooks, Add `ReactTestRenderer.act()` and `ReactTestUtils.act()` for batching updates, Support synchronous thenables passed to React.lazy(), Improve useReducer Hook lazy initialization API.
16.8.6	27 March 2019	Fix an incorrect bailout in useReducer(), Fix iframe warnings in Safari DevTools, Warn if contextType is set to Context.Consumer instead of Context, Warn if contextType is set to invalid values.
16.9.0	9 August 2019	Add `React.Profiler` API for gathering performance measurements programmatically. Remove unstable_ConcurrentMode in favor of unstable_createRoot
16.10.0	27 September 2019	Fix edge case where a hook update wasn't being memoized. Fix heuristic for determining when to hydrate, so we don't incorrectly hydrate during an update. Clear additional fiber fields during unmount to save memory. Fix bug with required text fields in Firefox. Prefer Object.is instead of inline polyfill, when available. Fix bug when mixing Suspense and error handling.
16.10.1	28 September	Fix regression in Next.js apps by allowing Suspense mismatch during hydration to silently proceed

	2019	
16.10.2	3 October 2019	Fix regression in react-native-web by restoring order of arguments in event plugin extractors
16.11.0	22 October 2019	Fix mouseenter handlers from firing twice inside nested React containers. Remove unstable_createRoot and unstable_createSyncRoot experimental APIs. (These are available in the Experimental channel as createRoot and createSyncRoot.)
16.12.0	14 November 2019	React DOM - Fix passive effects (`useEffect`) not being fired in a multi-root app. React Is - Fix `lazy` and `memo` types considered elements instead of components
16.13.0	26 February 2020	Features added in React Concurrent mode. Fix regressions in React core library and React Dom.
16.13.1	19 March 2020	Fix bug in legacy mode Suspense. Revert warning for cross-component updates that happen inside class render lifecycles
16.14.0	14 October 2020	Add support for the new JSX transform.
17.0.0	20 October 2020	"No New Features" enables gradual React updates from older versions. Add new JSX Transform, Changes to Event Delegation
17.0.1	22 October 2020	React DOM - Fixes a crash in IE11
17.0.2	22 March 2021	React DOM - Remove an unused dependency to address the `SharedArrayBuffer` cross-origin isolation warning.

Licensing

The initial public release of React in May 2013 used the Apache License 2.0. In October 2014, React 0.12.0 replaced this with the 3-clause BSD license and added a separate PATENTS text file that permits usage of any Facebook patents related to the software:[41]

> The license granted hereunder will terminate, automatically and without notice, for anyone that makes any claim (including by filing any lawsuit, assertion or other action) alleging (a) direct, indirect, or contributory infringement or inducement to infringe any patent: (i) by Facebook or any of its subsidiaries or affiliates, whether or not such claim is related to the Software, (ii) by any party if such claim arises in whole or in part from any software, product or service of Facebook or any of its subsidiaries or affiliates, whether or not such claim is related to the Software, or (iii) by any party relating to the Software; or (b) that any right in any patent claim of Facebook is invalid or unenforceable.

This unconventional clause caused some controversy and debate in the React user community, because it could be interpreted to empower Facebook to revoke the license in many scenarios, for example, if Facebook sues the licensee prompting them to take "other action" by publishing the

action on a blog or elsewhere. Many expressed concerns that Facebook could unfairly exploit the termination clause or that integrating React into a product might complicate a startup company's future acquisition.[42]

Based on community feedback, Facebook updated the patent grant in April 2015 to be less ambiguous and more permissive:[43]

> The license granted hereunder will terminate, automatically and without notice, if you (or any of your subsidiaries, corporate affiliates or agents) initiate directly or indirectly, or take a direct financial interest in, any Patent Assertion: (i) against Facebook or any of its subsidiaries or corporate affiliates, (ii) against any party if such Patent Assertion arises in whole or in part from any software, technology, product or service of Facebook or any of its subsidiaries or corporate affiliates, or (iii) against any party relating to the Software. [...] A "Patent Assertion" is any lawsuit or other action alleging direct, indirect, or contributory infringement or inducement to infringe any patent, including a cross-claim or counterclaim.[44]

The Apache Software Foundation considered this licensing arrangement to be incompatible with its licensing policies, as it "passes along risk to downstream consumers of our software imbalanced in favor of the licensor, not the licensee, thereby violating our Apache legal policy of being a universal donor", and "are not a subset of those found in the [Apache License 2.0], and they cannot be sublicensed as [Apache License 2.0]".[45] In August 2017, Facebook dismissed the Apache Foundation's downstream concerns and refused to reconsider their license.[46][47] The following month, WordPress decided to switch its Gutenberg and Calypso projects away from React.[48]

On September 23, 2017, Facebook announced that the following week, it would re-license Flow, Jest, React, and Immutable.js under a standard MIT License; the company stated that React was "the foundation of a broad ecosystem of open source software for the web", and that they did not want to "hold back forward progress for nontechnical reasons".[49]

On September 26, 2017, React 16.0.0 was released with the MIT license.[50] The MIT license change has also been backported to the 15.x release line with React 15.6.2.[51]

See also

- React Native
- Angular (web framework)
- Backbone.js
- Ember.js
- Next.js
- Svelte
- Vue.js
- Comparison of JavaScript libraries
- Web Components

References

1. Occhino, Tom; Walke, Jordan. "JS Apps at Facebook" (https://www.youtube.com/watch?v=GW0rj4sNH2w). *YouTube*. Retrieved 22 Oct 2018.

2. "Release 17.0.2" (https://github.com/facebook/react/releases/tag/v17.0.2). Retrieved 10 April 2021.
3. "React - A JavaScript library for building user interfaces" (https://reactjs.org). *React*. Retrieved 7 April 2018.
4. Krill, Paul (May 15, 2014). "React: Making faster, smoother UIs for data-driven Web apps" (https://www.infoworld.com/article/2608181/javascript/react--making-faster--smoother-uis-for-data-driven-web-apps.html). *InfoWorld*.
5. Hemel, Zef (June 3, 2013). "Facebook's React JavaScript User Interfaces Library Receives Mixed Reviews" (https://www.infoq.com/news/2013/06/facebook-react). *InfoQ*.
6. Dawson, Chris (July 25, 2014). "JavaScript's History and How it Led To ReactJS" (https://thenewstack.io/javascripts-history-and-how-it-led-to-reactjs/). *The New Stack*.
7. Dere, Mohan (2018-02-19). "How to integrate create-react-app with all the libraries you need to make a great app" (https://medium.freecodecamp.org/integrating-create-react-app-redux-react-router-redux-observable-bootstrap-altogether-216db97e89a3). *freeCodeCamp*. Retrieved 2018-06-14.
8. "Components and Props" (https://reactjs.org/docs/components-and-props.html#props-are-read-only). *React*. Facebook. Retrieved 7 April 2018.
9. "Refs and the DOM" (https://reactjs.org/docs/refs-and-the-dom.html). *React Blog*.
10. "React: The Virtual DOM" (https://www.codecademy.com/articles/react-virtual-dom). *Codecademy*. Retrieved 2021-10-14.
11. "Draft: JSX Specification" (https://facebook.github.io/jsx/). *JSX*. Facebook. Retrieved 7 April 2018.
12. "Why did we build React? – React Blog" (https://facebook.github.io/react/blog/2013/06/05/why-react.html).
13. "PayPal Isomorphic React" (https://medium.com/paypal-engineering/isomorphic-react-apps-with-react-engine-17dae662379c). Archived (https://web.archive.org/web/20190208124143/https://www.paypal-engineering.com/2015/04/27/isomorphic-react-apps-with-react-engine/) from the original on 2019-02-08.
14. "Netflix Isomorphic React" (http://techblog.netflix.com/2015/01/netflix-likes-react.html).
15. "Hooks at a Glance – React" (https://reactjs.org/docs/hooks-overview.html). *reactjs.org*. Retrieved 2019-08-08.
16. "What the Heck is React Hooks?" (https://blog.soshace.com/what-the-heck-is-react-hooks/). *Soshace*. 2020-01-16. Retrieved 2020-01-24.
17. "Using the State Hook – React" (https://reactjs.org/docs/hooks-state.html). *reactjs.org*. Retrieved 2020-01-24.
18. "Using the Effect Hook – React" (https://reactjs.org/docs/hooks-effect.html). *reactjs.org*. Retrieved 2020-01-24.
19. "Hooks API Reference – React" (https://reactjs.org/docs/hooks-reference.html). *reactjs.org*. Retrieved 2020-01-24.
20. "Rules of Hooks – React" (https://reactjs.org/docs/hooks-rules.html). *reactjs.org*. Retrieved 2020-01-24.
21. "Building Your Own Hooks – React" (https://reactjs.org/docs/hooks-custom.html). *reactjs.org*. Retrieved 2020-01-24.
22. "In Depth OverView" (https://facebook.github.io/flux/docs/in-depth-overview). *Flux*. Facebook. Retrieved 7 April 2018.
23. "Flux Release 4.0" (https://github.com/facebook/flux/releases/tag/4.0.0). *Github*. Retrieved 26 February 2021.
24. Johnson, Nicholas. "Introduction to Flux - React Exercise" (http://nicholasjohnson.com/react/course/exercises/flux/). *Nicholas Johnson*. Retrieved 7 April 2018.

25. Abramov, Dan. "The History of React and Flux with Dan Abramov" (http://threedevsandamaybe.com/the-history-of-react-and-flux-with-dan-abramov/). *Three Devs and a Maybe*. Retrieved 7 April 2018.
26. "State Management Tools - Results" (http://2016.stateofjs.com/2016/statemanagement/). *The State of JavaScript*. Retrieved 29 October 2021.
27. "Meeting Notes" (https://discuss.reactjs.org/c/meeting-notes). *React Discuss*. Retrieved 2015-12-13.
28. "reactjs/react-future - The Future of React" (https://github.com/reactjs/react-future). *GitHub*. Retrieved 2015-12-13.
29. "facebook/react - Feature request issues" (https://github.com/facebook/react/labels/Type:%20Feature%20Request). *GitHub*. Retrieved 2015-12-13.
30. Walke, Jordan. "FaxJS" (https://github.com/jordwalke/FaxJs). Retrieved 11 July 2019.
31. Papp, Andrea (4 April 2018). "The History of React.js on a Timeline" (https://blog.risingstack.com/the-history-of-react-js-on-a-timeline/). *RisingStack*. Retrieved 11 July 2019.
32. "Pete Hunt at TXJS" (https://www.youtube.com/watch?v=A0Kj49z6WdM).
33. Lardinois, Frederic (18 April 2017). "Facebook announces React Fiber, a rewrite of its React library" (https://techcrunch.com/2017/04/18/facebook-announces-react-fiber-a-rewrite-of-its-react-framework/). TechCrunch. Retrieved 19 April 2017.
34. "React Fiber Architecture" (https://github.com/acdlite/react-fiber-architecture). *Github*. Retrieved 19 April 2017.
35. "Facebook announces React Fiber, a rewrite of its React framework" (https://techcrunch.com/2017/04/18/facebook-announces-react-fiber-a-rewrite-of-its-react-framework/). *TechCrunch*. Retrieved 2018-10-19.
36. "GitHub - acdlite/react-fiber-architecture: A description of React's new core algorithm, React Fiber" (https://github.com/acdlite/react-fiber-architecture). *github.com*. Retrieved 2018-10-19.
37. "React v16.0" (https://reactjs.org/blog/2017/09/26/react-v16.0.html). react.js. 2017-09-26. Retrieved 2019-05-20.
38. "React v16.8" (https://reactjs.org/blog/2017/09/26/react-v16.0.html). react.js. 2019-02-16. Retrieved 2019-05-20.
39. "Introducing Hooks" (https://reactjs.org/docs/hooks-intro.html). react.js. Retrieved 2019-05-20.
40. url=https://reactjs.org/blog/2020/08/10/react-v17-rc.html
41. "React CHANGELOG.md" (https://github.com/facebook/react/blob/master/CHANGELOG.md#0120-october-28-2014). *GitHub*.
42. Liu, Austin. "A compelling reason not to use ReactJS" (https://medium.com/bits-and-pixels/a-compelling-reason-not-to-use-reactjs-beac24402f7b). *Medium*.
43. "Updating Our Open Source Patent Grant" (https://code.facebook.com/posts/1639473982937255/updating-our-open-source-patent-grant/).
44. "Additional Grant of Patent Rights Version 2" (https://github.com/facebook/react/blob/b8ba8c83f318b84e42933f6928f231dc0918f864/PATENTS). *GitHub*.
45. "ASF Legal Previously Asked Questions" (https://www.apache.org/legal/resolved.html). Apache Software Foundation. Retrieved 2017-07-16.
46. "Explaining React's License" (https://code.facebook.com/posts/112130496157735/explaining-react-s-license/). *Facebook*. Retrieved 2017-08-18.
47. "Consider re-licensing to AL v2.0, as RocksDB has just done" (https://github.com/facebook/react/issues/10191#issuecomment-323486580). *Github*. Retrieved 2017-08-18.
48. "WordPress to ditch React library over Facebook patent clause risk" (https://techcrunch.com/2017/09/15/wordpress-to-ditch-react-library-over-facebook-patent-clause-risk/). *TechCrunch*. Retrieved 2017-09-16.
49. "Relicensing React, Jest, Flow, and Immutable.js" (https://code.facebook.com/posts/300798627056246/relicensing-react-jest-flow-and-immutable-js/). *Facebook Code*. 2017-09-23.

50. Clark, Andrew (September 26, 2017). "React v16.0§MIT licensed" (https://reactjs.org/blog/2017/09/26/react-v16.0.html#mit-licensed). *React Blog*.
51. Hunzaker, Nathan (September 25, 2017). "React v15.6.2" (https://reactjs.org/blog/2017/09/25/react-v15.6.2.html). *React Blog*.

External links

- Official website (https://reactjs.org/)