

QUICK START

Getting Started

Tutorial

GUIDES

Why React?

Displaying Data

 JSX in Depth

 JSX Gotchas

Interactivity and Dynamic UIs

Multiple Components

Reusable Components

Forms

Working With the Browser

 More About Refs

Tooling integration

Reference

Getting Started

JSFiddle

The easiest way to start hacking on React is using the following JSFiddle Hello Worlds

- [React JSFiddle](#)
- [React JSFiddle without JSX](#)

Starter Kit

Download the starter kit to get started.

[Download Starter Kit 0.4.1](#)

In the root directory of the starter kit, create a `helloworld.html` with the following contents.

Code

```
<!DOCTYPE html>
<html>
  <head>
    <script src="build/react.min.js"></script>
    <script src="build/JSXTransformer.js"></script>
  </head>
  <body>
    <div id="example"></div>
    <script type="text/jsx">
```

```
    <h1>Hello, world!</h1>,
    document.getElementById('example')
);
</script>
</body>
</html>
```

The XML syntax inside of JavaScript is called JSX; check out the [JSX syntax](#) to learn more about it. In order to translate it to vanilla JavaScript we use `<script type="text/jsx">` and include `JSXTransformer.js` to actually perform the transformation in the browser.

Separate File

Your React JSX file can live in a separate file. Create the following `src/helloworld.js`.

Code

```
/** @jsx React.DOM */
React.renderComponent(
  <h1>Hello, world!</h1>,
  document.getElementById('example')
);
```

Then reference it from `helloworld.html`:

Code

```
<script type="text/jsx" src="src/helloworld.js"></script>
```

Offline Transform

First install the command-line tools (requires `npm`):

Then, translate your `src/helloworld.js` file to plain JavaScript:

Code

```
jsx --watch src/ build/
```

The file `build/helloworld.js` is autogenerated whenever you make a change.

Code

```
/** @jsx React.DOM */
React.renderComponent(
  React.DOM.h1(null, 'Hello, world!'),
  document.getElementById('example')
);
```

Note:

The comment parser is very strict right now, in order for it to pick up the `@jsx` modifier, two conditions are required. The `@jsx` comment block must be the first comment on the file. The comment must start with `/**` (`/*` and `//` will not work). If the parser can't find the `@jsx` comment, it will output the file without transforming it.

Update your HTML file as below:

Code

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello React!</title>
    <script src="build/react.min.js"></script>
```

```
<body>
  <div id="example"></div>
  <script src="build/helloworld.js"></script>
</body>
</html>
```

Want CommonJS?

If you want to use React within a module system, [fork our repo](#), `npm install` and run `grunt`. A nice set of CommonJS modules will be generated. Our `jsx` build tool can be integrated into most packaging systems (not just CommonJS) quite easily.

Next Steps

Check out [the tutorial](#) and the other examples in the `/examples` directory to learn more. Good luck, and welcome!

[Next →](#)

6 comments

[Add a comment](#)



Amor Mia

Koo koo!!!

[Reply](#) · 1 · [Like](#) · [Follow Post](#) · August 1 at 4:14am



Vasupat Chantakeaw · โรงพยาบาลราชวิถี

Wow I Like IT.

