

# React Fiber 漫谈

- Virtual DOM
- 为什么需要Fiber
- 什么是Fiber
- Fiber带来哪些改变

# Behind JSX: createElement

```
<div id="root">test</div>
```

=

```
React.createElement(  
  "div", // type  
  { id: "root" }, // props  
  "test" // children  
)
```

=

```
{ type: "div", props: { id: "root", children: ["test"] } }
```

# createElement(type, [props], [...children])

```
import React, { Component } from 'react'

const A = () => (
  <B foo={ true } bar="abc">
    <h1>hello</h1>
    <p>Welcome</p>
  </B>
)

var A = function A() {
  return _react2.default.createElement(
    B,
    { foo: true, bar: "abc" },
    _react2.default.createElement(
      "h1",
      null,
      "hello"
    ),
    _react2.default.createElement(
      "p",
      null,
      "Welcome"
    )
  );
};
```

Try it yourself: <https://babeljs.io/repl/>

# Host Component & Composite Component

	Host	Composite
首字母	小写	大写
element. type	String	class/function
定义	Renderer eg: ReactDOM	开发者

# StackReconciler

It's all about recursive

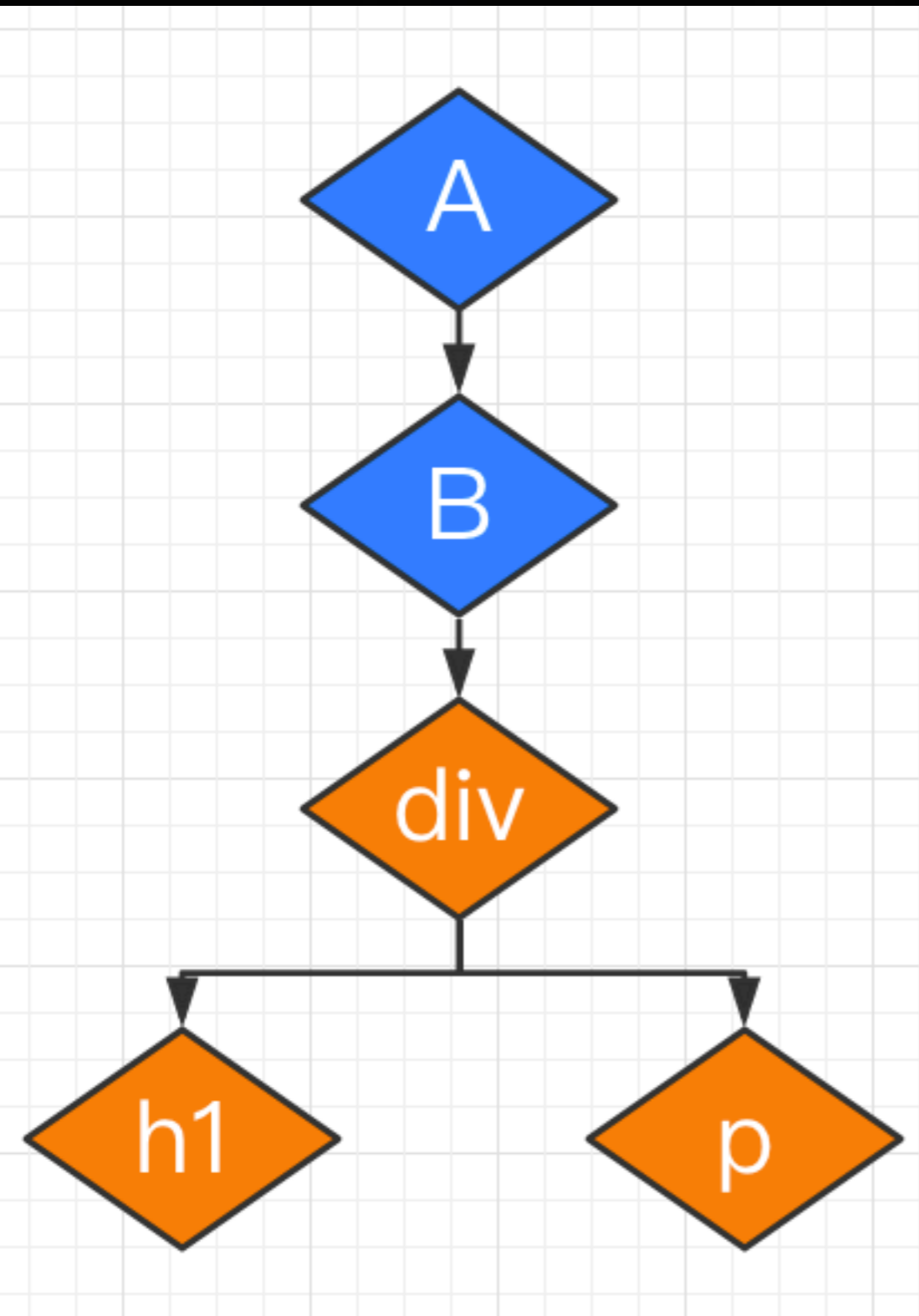
- 为什么不支持return数组?
- 为什么不支持stream渲染?
- 动画为什么容易卡顿?

# How StackReconciler works

```
const A = () => (  
  <B foo={ true } bar="abc">  
    <h1>hello</h1>  
    <p>Welcome</p>  
  </B>  
)  
  
const B = (props) => (  
  <div title={props.bar}>  
    {  
      props.foo  
      && props.children  
    }  
  </div>  
)  
ReactDOM.render(<A/>)
```

- mount A(null) =
- mount B({foo, bar, children}) =
- mount div
- document.createElement/  
setAttribute — 立即执行副作用
- appendChild()





- 没有专门的数据结构来描述组件树
- CompositeComponent 返回一个Element，只能表达深度
- HostComponent遍历 children负责广度
- 递归的函数调用-同步、不可打断

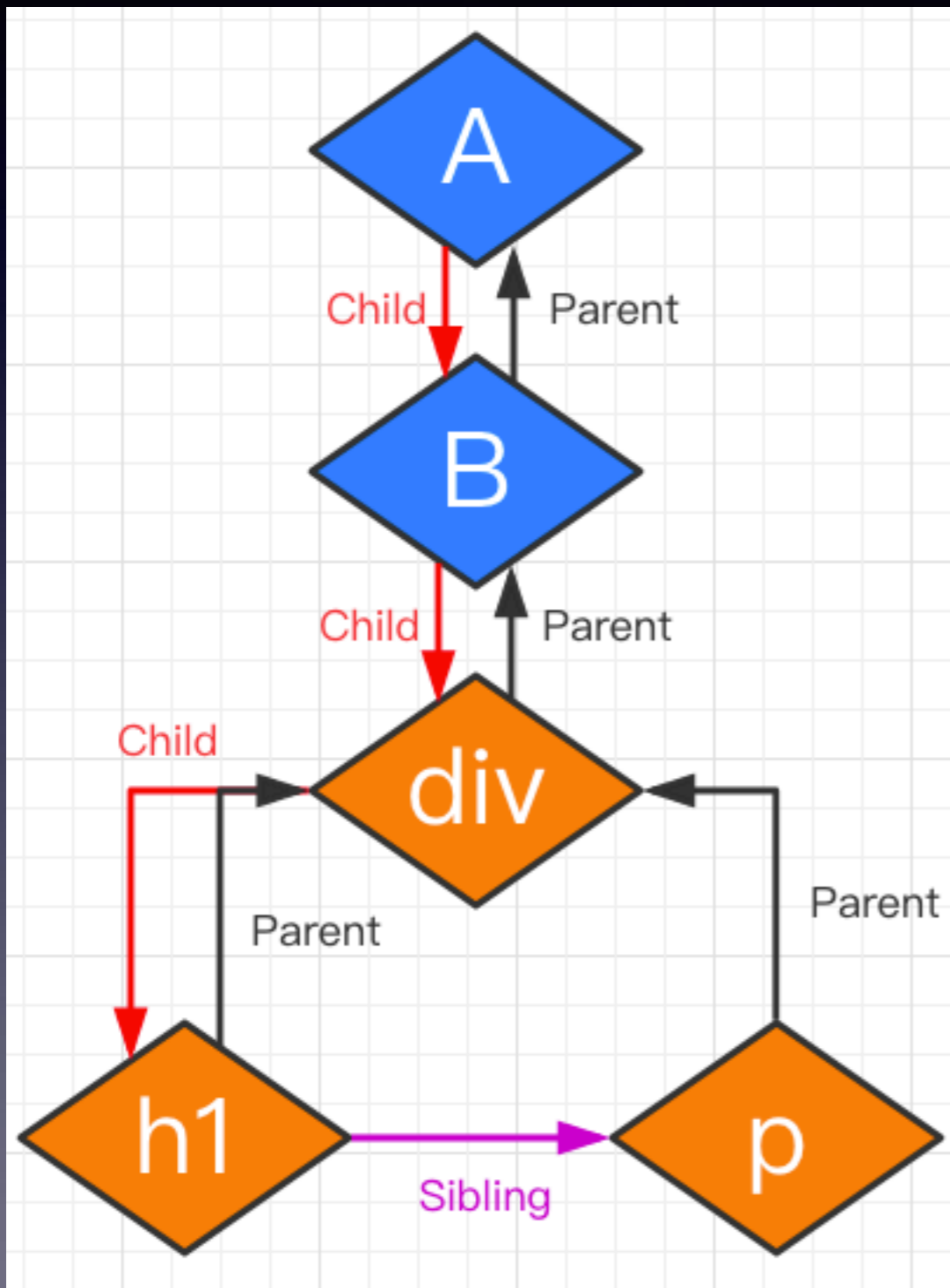
- 为什么不支持return数组? —— 递归只能返回一个元素
- 为什么不能stream? —— 函数同步执行不可中断
- 动画为什么容易卡顿? —— 占用主线程

Fiber - virtual call stack

“没有什么问题是增加一层抽象解决不了的，  
如果有，  
再加一层。”



# What is fiber



- 链表结构 - fiber
- while循环代替递归  
——可中断
- requestIdleCallback/  
requestAnimationFrame  
——可异步

- render / reconciliation(diff) : 收集side effect, 可中断
- commit : 应用side effect, 不可中断

- 更自由的写法
- 任务调度
- stream渲染

```
// render arrays
const ArrayItem = () => {
  return [
    <Foo/>,
    <Bar/>
  ]
}
```



```
// render Fragment
const FragmentItem = (item) => (
  <React.Fragment>
    <dt>{item.term}</dt>
    <dd>{item.description}</dd>
  </React.Fragment>
)

const Glossary = (props) => {
  return (
    <dl>
      {props.items.map(item => (
        <FragmentItem {...item}/>
      ))}
    </dl>
  );
}
```

# 基于优先级的任务调度

## 以及带来的问题

## 生命周期的顺序、次数难以保证

```
// low A
componentWillUpdate()
/* ----- */
// high B
componentWillUpdate()
componentDidUpdate()
/* ----- */
// restart low A
componentWillUpdate() // twice
componentDidUpdate()
```

## Render phase lifecycles

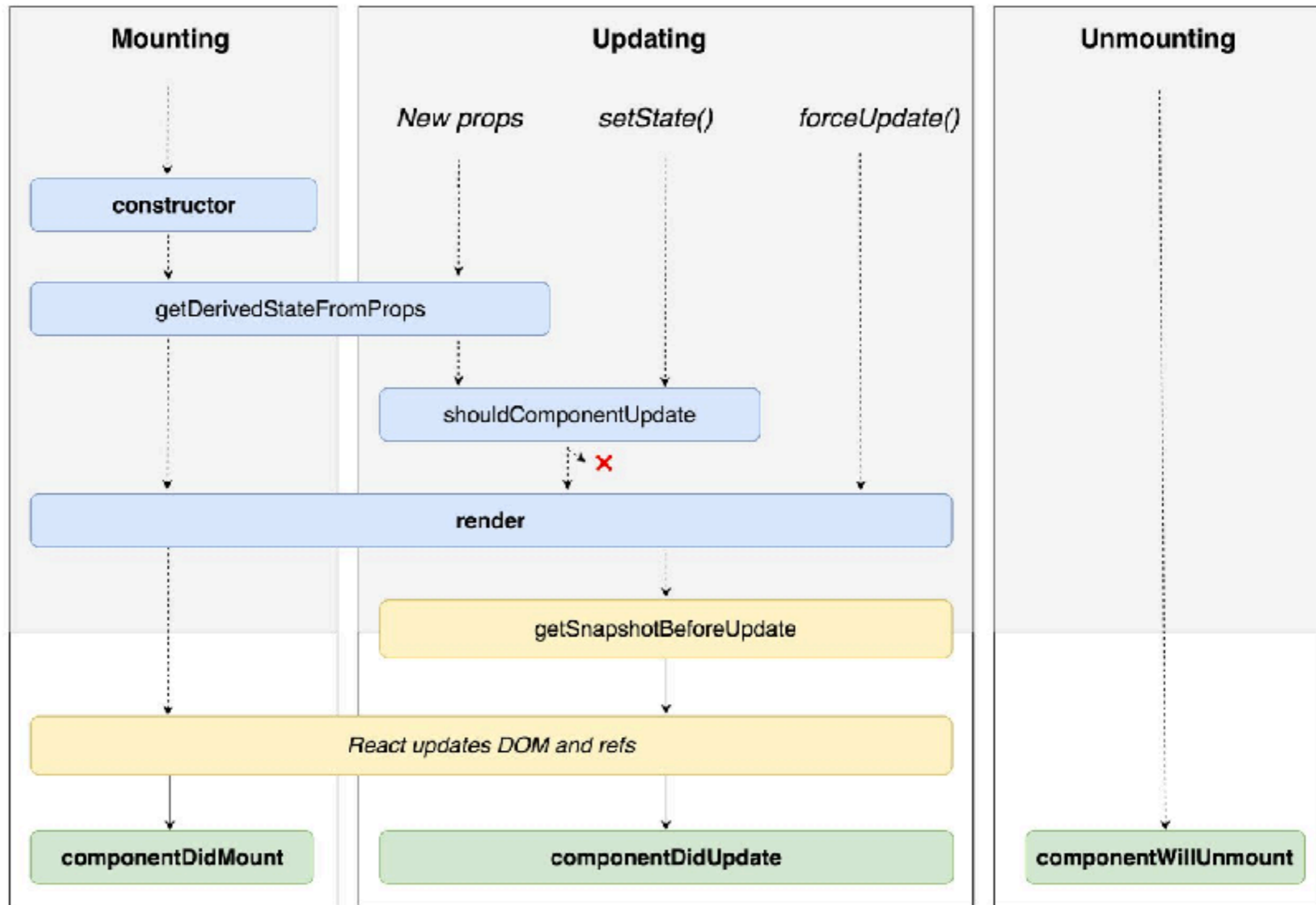
- `UNSAFE_componentWillMount`
- `UNSAFE_componentWillReceiveProps`
- `UNSAFE_componentWillUpdate`
- `shouldComponentUpdate`
- `static getDerivedStateFromProps` (new)

WARNING: No side effects!

**“Render Phase”**  
Pure and has no side effects.  
May be paused, aborted or  
restarted by React.

**“Pre-Commit Phase”**  
Can read the DOM.

**“Commit Phase”**  
Can work with DOM,  
run side effects,  
schedule updates.



# Scared?

- 目前为止没有开放异步能力
- 副作用放进 `componentDidMount/`  
`componentDidUpdate`中
- 短期内不会直接废弃 (v17也大概率会以  
`UNSAFE_`的方式支持)

# Other new features

- Context API
- Error Boundary

# React-Reconciler

<https://www.npmjs.com/package/react-reconciler>



```
* LICENSE file in the root directory of this source tree. An additional grant
* of patent rights can be found in the PATENTS file in the same directory.
*
*/
// Based off https://github.com/ForbesLindesay/umd/blob/master/template.js
;(function(f) {
  // CommonJS
  if (typeof exports === "object" && typeof module !== "undefined") {
    module.exports = f(require('react'));

    // RequireJS
  } else if (typeof define === "function" && define.amd) {
    define(['react'], f);

    // <script>
  } else {
    var g;
    if (typeof window !== "undefined") {
      g = window;
    } else if (typeof global !== "undefined") {
      g = global;
    } else if (typeof self !== "undefined") {
      g = self;
    } else {
      // works providing we're not in "use strict";
      // needed for Java 8 Nashorn
      // see https://github.com/facebook/react/issues/3037
      g = this;
    }
    g.ReactDOM = f(g.React);
  }
})(function(React) {
  return React.__SECRET_DOM_DO_NOT_USE_OR_YOU_WILL_BE_FIRED;
});
```



```
return React.__SECRET_DOM_DO_NOT_USE_OR_YOU_WILL_BE_FIRED;
```

# React like libraries

- Preact
- inferno
- react-lite
- Rax
- Anu

# Make your own React renderer with official Fiber-reconciler

<https://github.com/nitin42/Making-a-custom-React-renderer>

# Suspense

Beyond React v16