

React技术

高阶组件***

```
let Root = props => <div>{props.schoolName}</div>;
```

如果要在上例的Root组件进行增强怎么办？例如将Root组件的div外部再加入其它div。

```
let Wrapper = function (Component, props) {  
  return (  
    <div>  
      {props.schoolName}  
      <hr />  
      <Component />  
    </div>  
  );  
};  
  
let Root = props => <div>{props.schoolName}</div>;
```

为了以后用的方便，柯里化这个Wrapper函数。

```
import React from 'react';  
import ReactDOM from 'react-dom';  
  
let Wrapper = function (Component) {  
  function _wrapper(props) {  
    return (  
      <div>  
        {props.schoolName}  
        <hr />  
        <Component />  
      </div>  
    );  
  }  
  return _wrapper;  
};  
  
let Root = props => <div>Root</div>;  
  
let NewComp = Wrapper(Root) // 返回一个包装后的元素  
  
ReactDOM.render(<NewComp schoolName="magedu" />, document.getElementById('root'));
```

```
// 下面代码本身就是一个无状态组件，内部包裹这一个传入的组件，可以看做增强了传入的组件
// 传入的组件作为返回的新组件的子组件
function _wrapper(props) {
  return (
    <div>
      {props.schoolName}
      <hr />
      <Component />
    </div>
  );
}
// 那么上面代码的wrapper函数可以看做参数是一个组件，返回一个新组件，即高阶组件
```

简化Wrapper

```
let Wrapper = function (Component) {
  return (props) => {
    return (
      <div>
        {props.schoolName}
        <hr />
        <Component />
      </div>
    );
  };
};
```

再次变化

```
let Wrapper = function (Component) {
  return props =>
  (
    <div>
      {props.schoolName}
      <hr />
      <Component />
    </div>
  );
};
```

再次变化

```
import React from 'react';
import ReactDOM from 'react-dom';

let Wrapper = Component => props =>
  (<div>
    {props.schoolName}
```

```
    <hr />
    <Component />
  </div>);
```

```
let Root = props => <div>Root</div>;
```

```
let NewComp = Wrapper(Root)
```

```
ReactDOM.render(<NewComp schoolName="magedu" />, document.getElementById('root'));
```

装饰器

新版ES 2016中增加了装饰器的支持，因此可以使用装饰器来改造上面的代码。

```
@Wrapper // 这是不对的，装饰器装饰函数或类
let Root = props => <div>Root</div>;
```

ES 2016的装饰器只能装饰类，所以，将Root改写成类。

```
import React from 'react';
import ReactDOM from 'react-dom';
```

```
let Wrapper = Component => props =>
  (<div>
    {props.schoolName}
    <hr />
    <Component />
  </div>);
```

```
@Wrapper
class Root extends React.Component {
  render() {
    return <div>Root</div>;
  }
}
```

```
ReactDOM.render(<Root schoolName="magedu" />, document.getElementById('root'));
```

思考题

如何让Root也显示出schoolName?

```
import React from 'react';
import ReactDOM from 'react-dom';

let Wrapper = Component => props =>
  (<div>
    {props.schoolName}
```

```

    <hr />
    <Component {...props} />
  </div>);

@Wrapper
class Root extends React.Component {
  render() {
    return <div>{this.props.schoolName}</div>;
  }
}

ReactDOM.render(<Root schoolName="magedu" />, document.getElementById('root'));

```

使用 `<Component {...props} />` 相当于给组件增加了属性。

带参装饰器

想给装饰器函数增加一个id参数

```

import React from 'react';
import ReactDOM from 'react-dom';

// 带参装饰器函数
let Wrapper = id => Component => props =>
  (<div id = {id}>
    {props.schoolName}
    <hr />
    <Component {...props} />
  </div>);

@Wrapper('wrapper') // 带参
class Root extends React.Component {
  render() {
    return <div>{this.props.schoolName}</div>;
  }
}

ReactDOM.render(<Root schoolName="magedu" />, document.getElementById('root'));

```

通过上面的改写，就得到带参装饰器。

如果觉得不好接受，可以先写成原始的形式，熟悉了箭头函数语法后，再改成精简的形式。