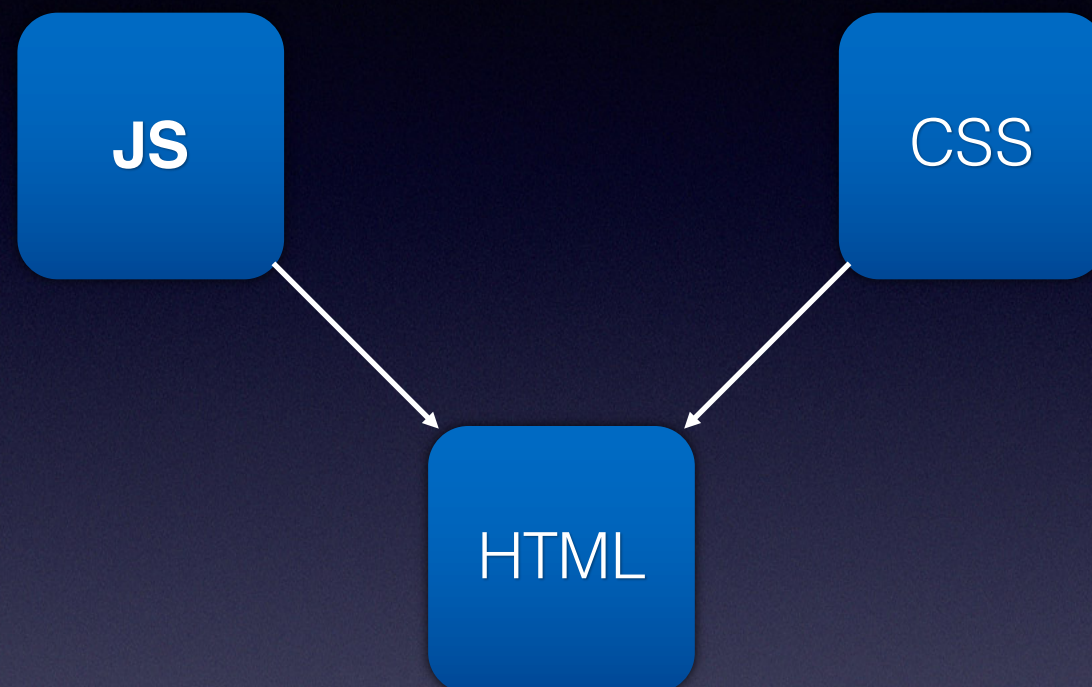


ReactJS分享

ReactJS简介

- 1、由facebook 开发
- 2、React为此引入了虚拟DOM (Virtual DOM) 的机制
- 3、单项数据流

完整的HTML结构



demo code:

```
<html>
<head>
<script type="text/javascript" src="js/demo.js">
<link href="css/demo.css" rel="stylesheet" type="text/css"/>
</head>
<body>
</body>
</html>
```


当前样式

code:

```
<html>
<head>
<script type="text/javascript">
    ***

</script>
<style type="text/css">
    *****

</style>
</head>
<body>
<script type="text/javascript">
    ***

</script>
</body>
</html>
```


setState用法

```
var LikeButton = React.createClass({
  getInitialState: function() {
    return {liked: false};
  },
  handleClick: function(event) {
    this.setState({liked: !this.state.liked});
  },
  render: function() {
    var text = this.state.liked ? 'like' : 'haven\'t liked';
    return (
      <p onClick={this.handleClick}>
        You {text} this. Click to toggle.
      </p>
    );
  }
});

React.render(
  <LikeButton />,
  document.getElementById('example')
);
```


state使用注意事项

- 组件其实是状态机 (State Machines)
- React 把用户界面当作简单状态机。把用户界面想像成拥有不同状态然后渲染这些状态，可以轻松让用户界面和数据保持一致。
- React 里，只需更新组件的 state，然后根据新的 state 重新渲染用户界面（不要操作 DOM）。React 来决定如何最高效地更新 DOM。
- State 工作原理
- 常用的通知 React 数据变化的方法是调用 `setState(data, callback)`。这个方法会合并 (merge) data 到 `this.state`，并重新渲染组件。渲染完成后，调用可选的 `callback` 回调。大部分情况下不需要提供 `callback`，因为 React 会负责把界面更新到最新状态。
- 补充：哪些组件应该有state ,哪些东西要作为state ,请参考官方介绍 <http://www.reactjs.cn/react/docs/interactivity-and-dynamic-uis.html>，并在实际应用中认真体会

传值：

```
render: function () {  
  if (this.state.hasData) {  
    var data = this.state.data;  
    return (  
      <div className="container-view">  
        <SearchBar bean={data.sb}/>  
        <SwiperView tbs={data.tbs}/>  
        <LocalProduct/>  
        <PanicBuyingView paincBuyingData={data.rup}/>  
        <CommercialList itemsData={data.as}/>  
        <FloorList ras={data.ras}/>  
        <MoreProductBar/>  
        <ProductWitsTagList/>  
        <LoadingLayout />  
      </div>  
    );  
  } else {  
    return (<div classaName="bottom_loading">  
      <LoadingLayout /></div>);  
  }  
}
```

```
var SearchBar = React.createClass({  
  
  componentDidMount: function () {  
    if (this.props.bean) {  
      var node = this.getDOMNode();  
      node.addEventListener('click', function () {  
        onClickEvent(this.props.bean.u, "EVENT_MALL_SEARCH_BAR_CLIEND", {});  
      }).bind(this);  
    }  
  },  
});
```


顶层API

- `React.createClass`
- `React.createElement`
- `React.render`
- `React.PropTypes`

组件API

- setState
- replaceState
- forceUpdate ()
- getDOMNode

组件常用方法

- render
- getInitialState
- getDefaultProps
- propTypes

组件生命周期

- `componentWillMount`
- `componentDidMount`
- `componentWillReceiveProps`
- `shouldComponentUpdate`
- `componentWillUpdate`
- `componentDidUpdate`
- `componentWillUnmount`