

极客学院
jikexueyuan.com

React 动画效果实现

React动画效果实现 — 课程概要

- 网页动画介绍
- CSS3动画介绍
- JS模拟动画介绍
- rAF动画介绍

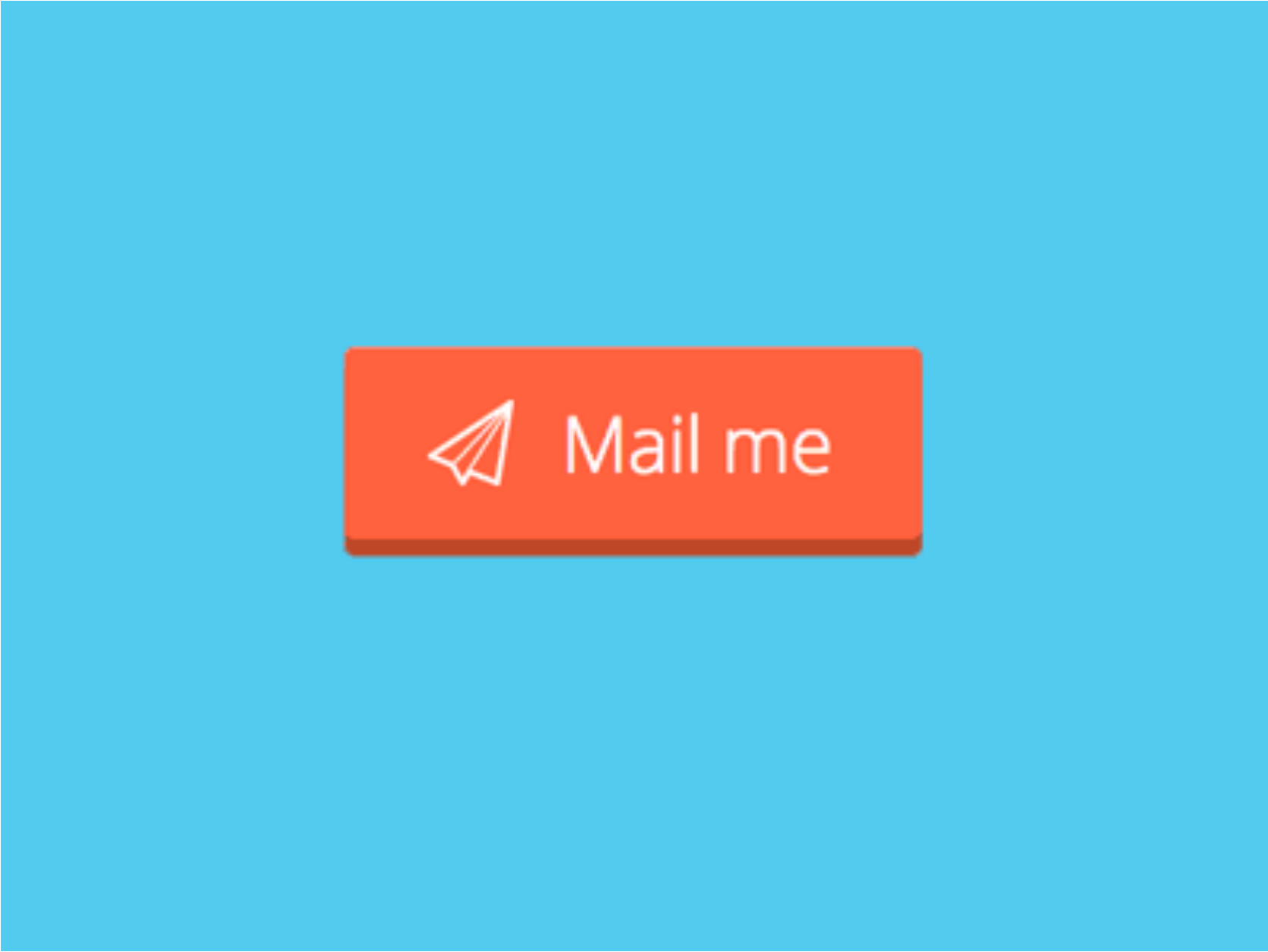


网页动画介绍

网页动画介绍

- 什么是网页动画
- 实现方法介绍
- 实现方法对比

■ 网页动画介绍 — 什么是网页动画



■ 网页动画介绍 — 什么是网页动画



网页动画介绍 — 实现方法介绍

CSS3: scale、transform、rotate、animation、transition

JS模拟: `$(...).animate()`、`$(...).show()`、`setInterval`、`setTimeout`

rAF (requestAnimationFrame) : 优化后的`setInterval`和`setTimeout`

SVG: `<animate>`

网页动画介绍 — 实现方法对比

名称	优点	缺点
CSS3动画	高性能、易于使用	兼容性差、无法实现复杂效果
JS模拟动画	兼容性强	性能差、难以调试
rAF动画	兼容性一般、性能较高	将被CSS3替代
SVG动画	高性能、能实现复杂效果	编写难度较大



CSS3动画介绍

CSS3动画介绍

- 什么是CSS3动画
- CSS3动画特点
- CSS3动画使用方法
- 实例演示

CSS3动画介绍 — 什么是CSS3动画

CSS3动画就是使用CSS3标准中的技术来绘制动画

HTML、CSS和JS都有标准，按照严格的标准制定流程进行制定

标准 ≠ 实现，标准由专家组制定，浏览器厂商不一定完全遵守标准

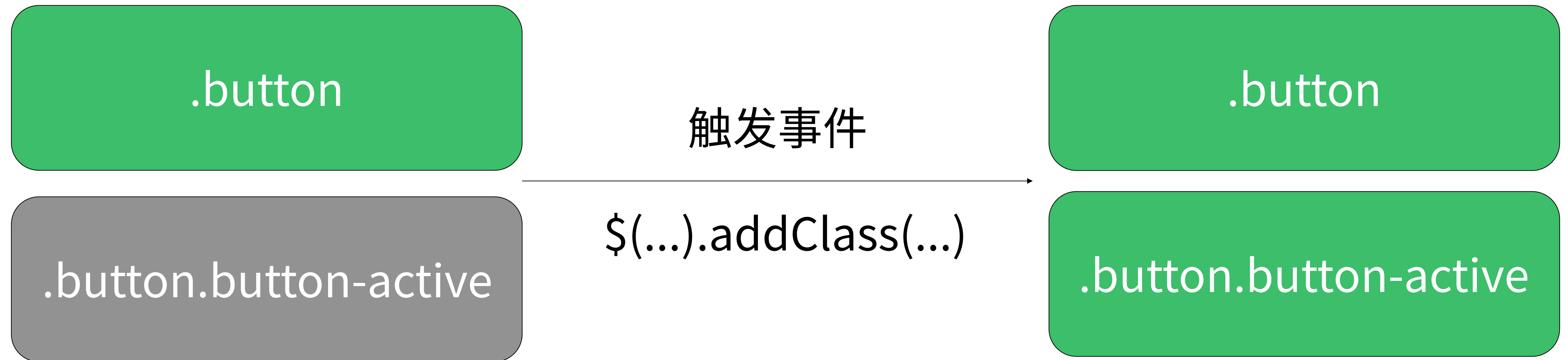
CSS3动画介绍 — CSS3动画特点

CSS3动画底层由浏览器实现具体的绘制流程，因此性能最好

CSS3动画能满足绝大部分交互的需求，上手简单，易于使用

但是，由于CSS3标准比较新，很多旧浏览器都无法兼容

CSS3动画介绍 — CSS3动画使用方法



常见的CSS3动画用法

CSS3动画介绍 — CSS3动画使用方法



React中CSS3动画用法

CSS3动画介绍 — CSS3动画使用方法

```
<ReactCSSTransitionGroup transitionName="example">
  {items}
</ReactCSSTransitionGroup>
```

```
.example-enter {
  opacity: 0.01;
  transition: opacity .5s ease-in;
}

.example-enter.example-enter-active {
  opacity: 1;
}
```

```
.example-leave {
  opacity: 1;
  transition: opacity .5s ease-in;
}

.example-leave.example-leave-active {
  opacity: 0.01;
}
```

React中CSS3动画用法

在React中使用CSS3动画

JS模拟动画介绍

JS模拟动画介绍

- 什么是JS动画
- JS动画特点
- JS动画使用方法
- 实例演示

JS模拟动画介绍 — 什么是JS模拟动画

为了兼容不支持CSS3的浏览器，用JS来模拟CSS3动画的效果

动画 = 连续播放的静止页面 ← 帧

transition: left 100px ease-in; → 总时间1秒，每秒向右移动1/100px

JS模拟动画介绍 — JS动画特点

兼容性强，适用于对浏览器兼容要求较高的项目

编写难度较大，通常不会手写代码，而是直接使用第三方库

性能需求大，如果改变属性过多，会占用多个浏览器渲染循环，导致卡顿

JS模拟动画介绍 — JS动画使用方法

```
function doMove() {  
    foo.style.left=(foo.style.left+10)+'px';  
    setTimeout(doMove,20);  
}
```

常见的JS模拟动画用法

JS模拟动画介绍 — JS动画使用方法

```
function doMove() {  
  foo.style.left=(foo.style.left+10)+'px';  
  setTimeout(doMove,20);  
  this.setState(...);  
}
```

React中JS模拟动画用法

JS模拟动画介绍 — 实例演示

在React中使用JS模拟动画



rAF动画介绍

rAF动画介绍

- 什么是rAF动画
- rAF动画特点
- rAF动画使用方法
- 实例演示

rAF动画介绍 — 什么是rAF动画

rAF = requestAnimationFrame

优化过的JS模拟动画

由于JS动画使用广泛但是性能差，因此提出rAF标准，用于优化性能

rAF动画介绍 — 什么是rAF动画

```
foo.style.width = ...  
foo.style.height = ...
```

改变width

改变height

普通JS模拟动画原理

rAF动画介绍 — 什么是rAF动画

```
foo.style.width = ...  
foo.style.height = ...
```

改变width
改变height

rAF原理

rAF动画介绍 — rAF动画特点

- 兼容性一般，IE 10+
- 性能较强，但是具体优化效果取决于浏览器
- 编写难度稍大，需要将动画转换成具体函数

rAF动画介绍 — rAF动画使用方法

```
1 var start = null;
2 var element = document.getElementById("SomeElementYouWantToAnimate");
3
4 function step(timestamp) {
5   if (!start) start = timestamp;
6   var progress = timestamp - start;
7   element.style.left = Math.min(progress/10, 200) + "px";
8   if (progress < 2000) {
9     window.requestAnimationFrame(step);
10  }
11 }
12
13 window.requestAnimationFrame(step);
```

常见的rAF动画用法

rAF动画介绍 — rAF动画使用方法

```
1  resolveAnimationFrame: function() {  
2    var timestamp = new Date();  
3    var timeRemaining = Math.max(0, this.props.animationCompleteTimestamp - timestamp);  
4    if (timeRemaining > 0) {  
5      this.setState({position: timeRemaining});  
6    }  
7  },  
8  componentWillUpdate: function() {  
9    if (this.props.animationCompleteTimestamp) {  
10     requestAnimationFrame(this.resolveAnimationFrame);  
11    }  
12  }
```

React中rAF动画用法

在React中使用rAF动画

React动画效果实现

本课程中我们学习了React中动画的实现方式，你应当掌握了以下知识：

- 什么是动画效果
- CSS3动画效果实现
- JS模拟动画效果实现
- rAF动画效果实现

你可以使用这些技巧来编写更具交互性、更符合React特性的组件，如果想继续提高，你可以在极客学院继续学习React系列课程。

极客学院

jikexueyuan.com

中国最大的IT职业在线教育平台

