

# Web APIs 第四天

DOM- 事件高级



# 目录

Contents

- ◆ 事件对象
- ◆ 事件流
- ◆ 事件委托
- ◆ 综合案例

# 学习目标

Learning Objectives

1. 掌握事件对象的使用
2. 具备利用操作数据渲染学生信息案例的能力



# 事件对象

- 获取事件对象
- 事件对象常用属性

## 1.1 获取事件对象

目标：能说出什么是事件对象

- 事件对象是什么

- 也是个对象，这个对象里有事件触发时的相关信息
- 例如：鼠标点击事件中，事件对象就存了鼠标点在哪个位置等信息

- 如何获取

- 在事件绑定的回调函数的第一个参数就是事件对象
- 一般命名为event、ev、e

```
元素.addEventListener('click', function (e) {  
  
})
```

事件对象



## 总结

1. 事件对象是什么?
  - 也是个对象，这个对象里有事件触发时的相关信息
2. 事件对象在哪里?
  - 在事件绑定的回调函数的第一个参数就是事件对象

```
元素.addEventListener('click', function (e) {  
  
})
```



# 事件对象

- 获取事件对象
- 事件对象常用属性

## 1.1 获取事件对象

目标：能够使用常见事件对象属性

- 部分常用属性

- type

- 获取当前的事件类型

- clientX/clientY

- 获取光标相对于浏览器可见窗口左上角的位置

- offsetX/offsetY

- 获取光标相对于当前DOM元素左上角的位置

- key

- 用户按下的键盘键的值

- 现在不提倡使用keyCode



## 案例

### 跟随鼠标案例

需求：一张图片一直跟着鼠标移动

分析：

- ①：鼠标在页面中移动，用到 mousemove 事件
- ②：不断把鼠标在页面中的坐标位置给图片left和top值即可



## 案例

### 按下回车发布微博案例

需求：按下回车键盘，可以发布信息

分析：

- ①：用到按下键盘事件 `keydown` 或者 `keyup` 都可以
- ②：如果用户按下的是回车键盘，则发布信息
- ③：按下键盘发布新闻，其实和点击发布按钮效果一致 `send.click()`



# 目录

Contents

- ◆ 事件对象
- ◆ 事件流
- ◆ 事件委托
- ◆ 综合案例



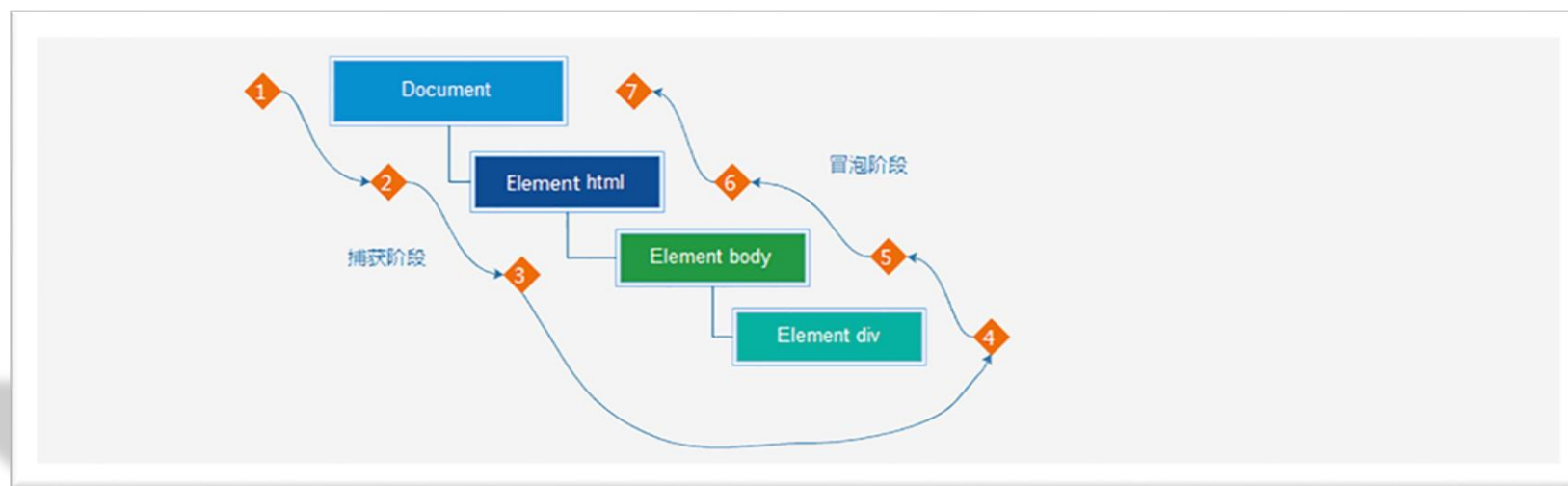
## 事件流

- 事件流与两个阶段说明
- 事件捕获和事件冒泡
- 阻止事件流动

## 2.1 事件流和两个阶段说明

目标：能够说出事件流经过的2个阶段

- **事件流**指的是事件完整执行过程中的流动路径



- 说明：假设页面里有个div，当触发事件时，会经历两个阶段，分别是捕获阶段、冒泡阶段
- 简单来说：捕获阶段是 从父到子 冒泡阶段是从子到父



## 事件流

- 事件流与两个阶段说明
- 事件捕获和事件冒泡
- 阻止事件流动

## 2.2 事件捕获和事件冒泡

目标：能够说出事件流经过的2个阶段

- **事件冒泡概念：**

当一个元素的事件被触发时，同样的事件将会在该元素的所有祖先元素中依次被触发。这一过程被称为事件冒泡

- 简单理解：当一个元素触发事件后，会依次向上调用所有父级元素的同名事件
- 事件冒泡是默认存在的

## 2.2 事件捕获和事件冒泡

目标：能够说出事件流经过的2个阶段

- 事件捕获概念：

从DOM的根元素开始去执行对应的事件 (从外到里)

- 事件捕获需要写对应代码才能看到效果

- 代码：

```
DOM.addEventListener(事件类型, 事件处理函数, 是否使用捕获机制)
```

- 说明：

- addEventListener第三个参数传入true代表是捕获阶段触发（很少使用）
- 若传入false代表冒泡阶段触发，默认就是false
- 若是用 L0 事件监听，则只有冒泡阶段，没有捕获





## 事件流

- 事件流与两个阶段说明
- 事件捕获和事件冒泡
- 阻止事件流动

## 2.3 阻止事件流动

目标：能够写出阻止事件流动的代码

- 因为默认就有冒泡模式的存在，所以容易导致事件影响到父级元素
- 若想把事件就限制在当前元素内，就需要阻止事件流动
- 阻止事件流动需要拿到事件对象
- 语法：

```
事件对象.stopPropagation()
```

- 此方法可以阻断事件流动传播，不光在冒泡阶段有效，捕获阶段也有效

## 2.3 阻止事件流动

目标：能够写出阻止事件流动的代码

- 鼠标经过事件：
  - mouseover 和 mouseout 会有冒泡效果
  - mouseenter 和 mouseleave 没有冒泡效果(推荐)

## 2.3 阻止事件流动

目标：能够写出阻止事件流动的代码

- 阻止默认行为，比如链接点击不跳转，表单域的跳转
- 语法：

```
e.preventDefault()
```

## 2.3 阻止事件流动

目标：能够写出阻止事件流动的代码

- 两种注册事件的区别：

- 传统on注册（L0）

- 同一个对象,后面注册的事件会覆盖前面注册(同一个事件)
- 直接使用null覆盖偶就可以实现事件的解绑
- 都是冒泡阶段执行的

- 事件监听注册（L2）

- 语法: `addEventListener(事件类型, 事件处理函数, 是否使用捕获)`
- 后面注册的事件不会覆盖前面注册的事件(同一个事件)
- 可以通过第三个参数去确定是在冒泡或者捕获阶段执行
- 必须使用 `removeEventListener(事件类型, 事件处理函数, 获取捕获或者冒泡阶段)`
- 匿名函数无法被解绑



# 目录

Contents

- ◆ 事件对象
- ◆ 事件流
- ◆ 事件委托
- ◆ 综合案例



## 事件委托

### 3. 事件委托

目标：能够说出事件委托的好处

- 事件委托是利用事件流的特征解决一些开发需求的知识技巧
- 总结：
  - **优点：**给父级元素加事件（可以提高性能）
  - **原理：**事件委托其实是利用事件冒泡的特点
  - **实现：**事件对象.target 可以获得真正触发事件的元素





# 目录

Contents

- ◆ 事件对象
- ◆ 事件流
- ◆ 事件委托
- ◆ 综合案例

## 案例

### 渲染学生信息案例

需求：点击录入按钮，可以增加学生信息

#### 新增学生信息

姓名:  年龄:  性别:  薪资:  就业城市:

#### 就业榜

学号	姓名	年龄	性别	薪资	就业城市	操作
1001	欧阳霸天	19	男	20000	上海	删除
1002	令狐霸天	29	男	30000	北京	删除
1003	诸葛霸天	39	男	2000	北京	删除

## 案例

### 渲染学生信息案例

需求：点击录入按钮，可以增加学生信息

说明：

本次案例主要目的是为了后面学习Vue做铺垫（数据驱动视图）

## 案例

### 渲染学生信息案例

需求：点击录入按钮，可以增加学生信息

分析：

1. 不管添加还是删除，都是操作的数据（数组），然后从新渲染页面

需求①：添加数据

点击录入按钮，把表单里面的值都放入数组里面

学号自动生成，是数组最后一个数据的学号+1

需求②：渲染

把数组的数据渲染到页面中，同时清空表单里面的值，下拉列表的值复原

注意，渲染之前，先清空以前渲染的内容

因为多次渲染，最好封装为函数

## 案例

### 渲染学生信息案例

需求：点击录入按钮，可以增加学生信息

分析：

1. 不管添加还是删除，都是操作的数据（数组），然后从新渲染页面

需求③：删除数据

为了提高性能，最好使用事件委托方式，找到点击的是链接 `e.target.tagName`

根据当前的删除链接，找到这条数据

需要会用 `findIndex` 方法找到数组元素所在的索引号方便删除

然后使用 `splice` 来删除对应数据

重新渲染



思考

1. 整理今天笔记
2. 练习生成表格案例
3. 作业

今天多一份拼搏，明日多一份欢笑



传智教育旗下高端IT教育品牌