

# 浏览器事件模型

# DOM 0级 事件模型

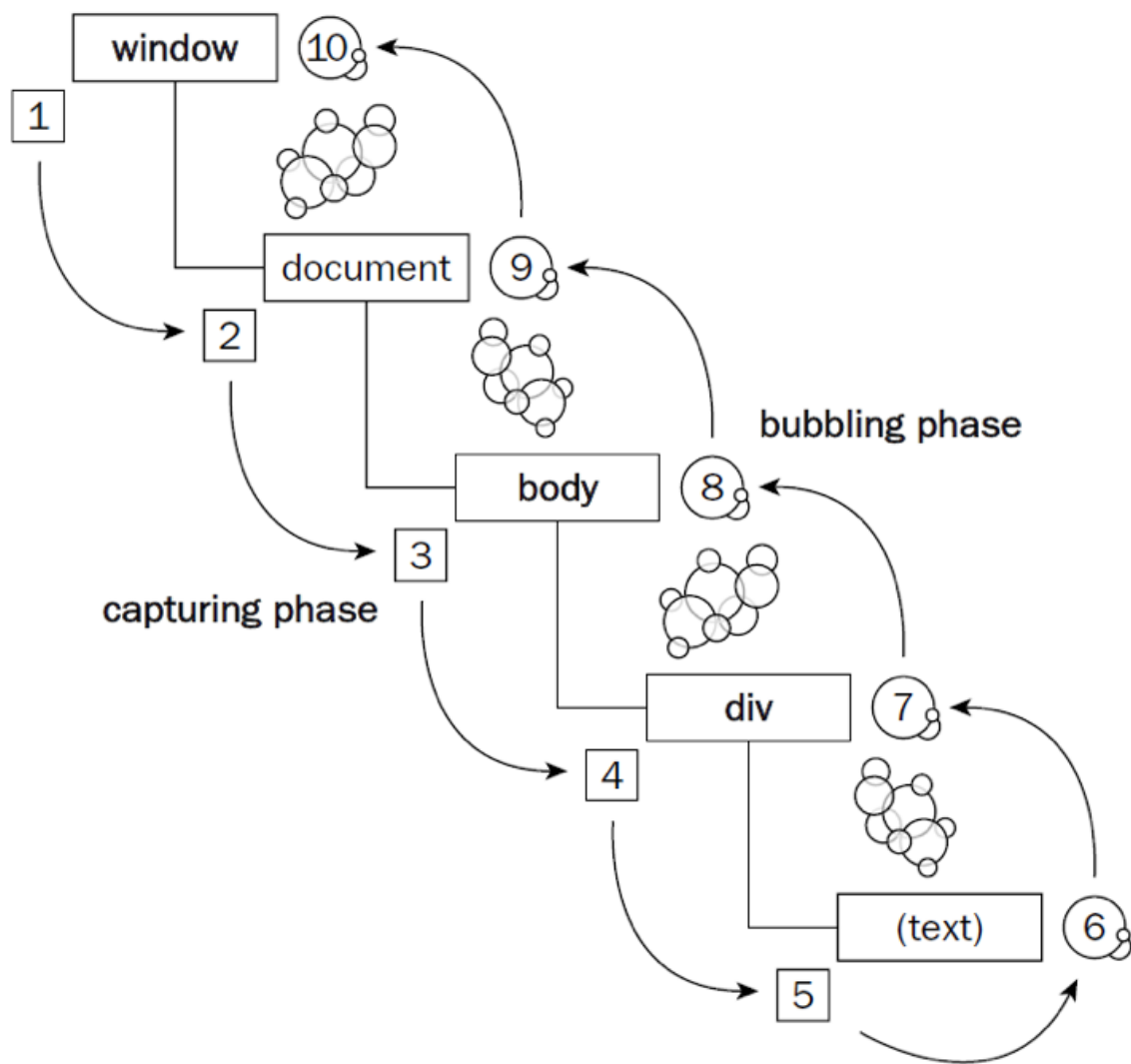
# DOM 0级 事件模型

- `<input type="button" onclick="doSomething()" />`
- `input.onclick = function() { ... }`
- `event = event || window.event;`
- `var target = event.target || event.srcElement;`
- ...

## DOM 2级 事件模型

# DOM 2级 事件模型

- **addEventListener(eventType, listener, useCapture)**
- **attachEvent(eventName, handler)**



# jQuery 事件模型

# jQuery 事件模型

- 提供了统一的事件处理方法
- 允许添加多个事件处理函数
- 使用标准的事件名称（不带on）
- 事件实例做为事件处理函数的第一个参数
- 标准化事件实例最常用的属性
- 提供了统一的事件取消和阻止默认行为的方法



# 添加事件处理

- `on(eventType[, selector][, data], handler)`

- ~~`bind()`~~、~~`delegate()`~~、~~`live()`~~

# 统一方法和属性

- 阻止冒泡：`stopPropagation()`
- 阻止默认行为：`preventDefault()`
- 阻止冒泡和默认行为：`return false`

# 所有支持的事件

- blur
- change
- click
- dblclick
- error
- focus
- focusin
- focusout
- keydown
- keypress
- keyup
- load
- unload

# 所有支持的事件

- mousedown
- mouseenter
- mouseleave
- mousemove
- mouseout
- mouseover
- mouseup
- ready
- resize
- scroll
- select
- submit

# 一次性的事件处理

- `one(eventType[, selector][, data], handler)`

# 移除事件处理

- `off(eventType[, selector][, handler])`
- `off()`

## 事件实例对象

# 事件实例对象的属性

**altKey**

**bubbles**

**button**

**cancelable**

**charCode**

**clientX**

**clientY**

**ctrlKey**

**currentTarget**

**data**

**detail**

**delegateTarget**

**eventPhase**

**metaKey**

**namespace**

**offsetX**

**offsetY**



# 事件实例对象的属性

**originalTarget**

**originalEvent**

**pageX**

**pageY**

**prevValue**

**relatedTarget**

**result**

**screenX**

**screenY**

**shiftKey**

**target**

**timeStamp**

**type**

**view**

**which**

# 事件实例对象的方法

- **preventDefault()**
- **stopPropagation()**
- **stopImmediatePropagation()**
- **isDefaultPrevented()**
- **isPropagationStopped()**
- **isImmediatePropagationStopped()**

# 触发事件

- **trigger(eventType[, data])**
- **triggerHandler(eventType[, data])**

## 区别

- **triggerHandler 相比 trigger :**
- **不会触发浏览器默认事件 ;**
- **不会产生事件冒泡 ;**
- **只触发jQuery对象集合中第一个元素的事件处理函数 ;**
- **返回的是事件处理函数的返回值 , 而不是jQuery对象。**

# 快捷方法

**blur**

**change**

**click**

**dblclick**

**focus**

**focusin**

**focusout**

**keydown**

**keypress**

**keyup**

# 快捷方法

**mousedown**

**mouseenter**

**mouseleave**

**mousemove**

**mouseout**

**mouseover**

**mouseup**

**ready**

**resize**

**scroll**

**select**

**submit**

# 快捷方法的使用

- **eventName([data,] handler)**
- **eventName()**

# hover 方法

- **hover(enterHandler, leaveHandler)**
- **hover(handler)**



# 自定义事件

- **on(customEvent)**
- **trigger(customEvent)**

# 事件命名空间

- `eventName.namespace`

# 总结