事件

从入门到工作2020版

版权声明

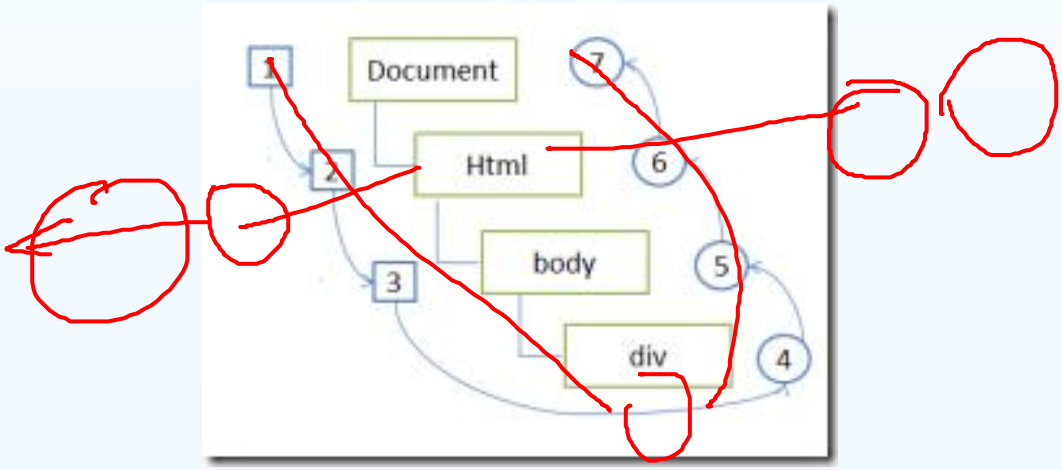
- 本内容版权归属杭州饥人谷教育科技有限公司所有。
- 任何媒体、网站、个人未经本公司授权不得转载、链接、转帖，或以其他方式复制、发布、发表。
- 已获得饥人谷授权的媒体、网站、个人在使用时必须注明“资料来源：饥人谷”
- 如咨询课程，微信联系xiedaimala02 或xiedaimala03 ，或者点击链接直接咨询https://dwz.cn/pPsjntxY

事件

- ✓ 事件是在编程时系统内发生的动作或者发生的事情
  - 单击
  - 双击
  - 鼠标放置
  - 表单内容发生变化
  - 拖拽
  - 页面滚动
  - 触发/失去焦点
  - 键盘按下
  - 提交表单
  - ……
- ✓ 事件全览
  - https://developer.mozilla.org/en-US/docs/Web/Events

DOM事件流

- 三个阶段
  - ✓ 事件捕获阶段，处于目标阶段，事件冒泡阶段



事件处理程序

- ✓ 当事件发生时执行的函数

- ✓ 三种绑定方法，范例中的click可以换成任意事件类型

```
<!-- 方法1，在html里绑定，尽量不用 -->
<div onclick="sayHello">点我</div>

<script>
<!-- 方法2，给dom元素的onclick赋值 -->
$btn.onclick = function(e) {
  console.log('点我')
}

<!-- 方法3，调用dom的addEventListener函数 -->
$btn.addEventListener('click', function(e) {
  console.log('点我')
}, true)
</script>
```

addEventListener

- target.addEventListener(type, listener, options)
  - type: 事件类型
  - listener: 事件处理方法
  - options: 可选，默认全是false。{capture: 是否捕获阶段监听, once: 是否只监听一次, passive: 是否忽略preventDefault }
- target.addEventListener(type, listener, useCapture)
  - useCapture: 可选，true表示在捕获阶段调用listener，false表示冒泡阶段处理
- target.removeEventListener
  - target.removeEventListener('click', handler)
  - target.removeEventListener('click', handler, true)

```
<input id="btn" type="button" value="Click Here" />
<script>
  let $btn = document.querySelector('#btn')
  $btn.addEventListener('click', function() {
    console.log('在捕获阶段监听，且点只执行一次')
  }, {capture: true, once: true})
</script>
```

冒泡与捕获

- ✓ 停止事件传播
  - e.stopPropagation()
  - 范例: http://js.jirengu.com/yifif

```
<div class="parent">container
  <div class="child">box
    <div class="target">target</div>
  </div>
</div>

<style>
  div {
    border: 1px solid;
    margin: 10px;
  }
</style>
```

```
const $ = s => document.querySelector(s)
$('.parent').addEventListener('click', function(e){
  console.log('在捕获阶段，parent 开始处理')
}, true)

$('.child').addEventListener('click', function(e){
  console.log('在捕获阶段，child 开始处理')
  //e.stopPropagation()
}, true)

$('.target').addEventListener('click', function(e){
  console.log('在捕获阶段，target 开始处理')
}, true)

$('.parent').addEventListener('click', function(e){
  console.log('在冒泡阶段，parent 开始处理')
}, false)

$('.child').addEventListener('click', function(e){
  console.log('在冒泡阶段，child 开始处理')
}, false)

$('.target').addEventListener('click', function(e){
  console.log('在冒泡阶段，target 开始处理')
}, false)
```

阻止默认事件

- e.preventDefault()
  - ✓ 链接不自动跳转
  - ✓ 表单不自动提交

- 与passive选项的关系
  - ✓ 移动端滚动性能优化

阻止默认事件

- 范例1

✓ 当用户点击链接时，如果是本站域名链接则跳转，否则不跳转 http://js.jirengu.com/koliv/

```
<a href="http://js.jirengu.com/safag/">站内链
接:js.jirengu.com/safag</a>
<a href="https://baidu.com">站外链接:baidu.com</a>

<script>
let $$a = document.querySelectorAll('a')
$$a.forEach($a => {
  $a.addEventListener('click', function(event) {
    let href = this.getAttribute('href')
    if(href.includes(location.host)) {
      location.href = href
    } else {
      event.preventDefault()
    }
  })
})
</script>
```

阻止默认事件

- 范例2
  - ✓ 表单提交前做校验 http://js.jirengu.com/xanih

```
<form action="/login" methods="GET">
  <input  id="username" type="text" name="username">
  <input type="submit">
  <span class="error"></span>
</form>

<script>
let $form = document.querySelector('form')
let $username = document.querySelector('#username')
let $error = document.querySelector('.error')
const validUsername = str=> /^\w{6,15}$/g.test(str)

$form.addEventListener('submit', function(e) {
  e.preventDefault()
  if(validUsername($username.value)) {
    this.submit()
  } else {
    $error.innerText = '密码不符合规则'
  }
})
</script>
```

事件代理

- ✓ 事件绑定代理给父元素，由父元素根据事件来源统一处理
- ✓ 适用于可能会新增子元素对场景
- ✓ 事件代理实际上是事件冒泡的应用

一个家庭成员管账的例子

为什么用事件代理

- 案例
  - http://js.jirengu.com/qubog/edit

```html
<div class="container">
  <div class="box">
    <span>box1</span>
  </div>
  <div class="box">
    <span>box2</span>
  </div>
  <div class="box">
    <span>box3</span>
  </div>
</div>
<button id="add">add</button>
<p id="log"></p>
```

```js
let $container = document.querySelector('.container')
let $$boxs = document.querySelectorAll('.box')
let $add = document.querySelector('#add')
let $log = document.querySelector('#log')

const createBox = text => {
  let $box = document.createElement('div')
  $box.classList.add('box')
  let $span = document.createElement('span')
  $span.innerText = text
  $box.appendChild($span)
  $container.appendChild($box)
}

let i = 4
$add.onclick = function() {
  createBox('box' + i++)
}

//这样用有什么问题
$$boxs.forEach($box => {
  $box.onclick = function(){
    $log.innerText = this.innerText
  }
})
```

使用事件代理

- 案例
  - http://js.jirengu.com/qubog/edit

```
<div class="container">
  <div class="box">
    <span>box1</span>
  </div>
  <div class="box">
    <span>box2</span>
  </div>
  <div class="box">
    <span>box3</span>
  </div>
</div>
<button id="add">add</button>
<p id="log"></p>
```

```
let $container = document.querySelector('.container')
let $$boxs = document.querySelectorAll('.box')
let $add = document.querySelector('#add')
let $log = document.querySelector('#log')

const createBox = text => {
  let $box = document.createElement('div')
  $box.classList.add('box')
  let $span = document.createElement('span')
  $span.innerText = text
  $box.appendChild($span)
  $container.appendChild($box)
}

let i = 4
$add.onclick = function() {
  createBox('box' + i++)
}

$container.onclick = function(e){
  let $box = e.path.find(el => el.classList
                        &&el.classList.contains('box'))
  if($box) {
    $log.innerText = $box.innerText
  }
}
```

常见事件范例

- 范例
- ✓ http://js.jirengu.com/kegeb/edit