

JavaScript DOM实战案例

王红元 coderwhy

目录

content



1 window定时器

2 轮播消息提示

3 关闭隐藏消息

4 侧边栏展示

5 tab切换实现

6 王者荣耀轮播图

7 常见的事件

window定时器方法

- 有时我们并不想立即执行一个函数，而是等待特定一段时间之后再执行，我们称之为“计划调用（scheduling a call）”。
- 目前有两种方式可以实现：
 - `setTimeout` 允许我们将函数推迟到一段时间间隔之后再执行。
 - `setInterval` 允许我们重复运行一个函数，从一段时间间隔之后开始运行，之后以该时间间隔连续重复运行该函数。
- 并且通常情况下有提供对应的取消方法：
 - `clearTimeout`：取消`setTimeout`的定时器；
 - `clearInterval`：取消`setInterval`的定时器；
- 大多数运行环境都有内置的调度程序，并且提供了这些方法：
 - 目前来讲，所有浏览器以及 `Node.js` 都支持这两个方法；
 - 所以我们后续学习Node的时候，也可以在Node中使用它们；

setTimeout的使用

■ setTimeout的语法如下：

```
let timerId = setTimeout(func|code, [delay], [arg1], [arg2], ...)
```

❑ **func|code**：想要执行的函数或代码字符串。

➤ 一般传入的都是函数，由于某些历史原因，支持传入代码字符串，但是不建议这样做；

❑ **delay**：执行前的延时，以毫秒为单位（1000 毫秒 = 1 秒），默认值是 0；

❑ **arg1, arg2...**：要传入被执行函数（或代码字符串）的参数列表；

■ clearTimeout方法：

❑ setTimeout 在调用时会返回一个 “**定时器标识符 (timer identifier)** ”，我们可以使用它来取消执行。

```
var timerID = setTimeout(function(name, age) {  
    console.log("定时器", name, age)  
}, 2000, "why", 18);  
  
clearTimeout(timerID)
```

setInterval的使用

■ setInterval 方法和 setTimeout 的语法相同：

```
let timerId = setInterval(func|code, [delay], [arg1], [arg2], ...)
```

- 所有参数的意义也是相同的；
- 不过与 setTimeout 只执行一次不同，setInterval 是每间隔给定的时间周期性执行；

■ clearInterval方法：

- setInterval也会返回一个 “定时器标识符 (timer identifier) ”，我们可以通过clearInterval来取消这个定时器。

```
var timerId = setInterval(function() {  
  console.log("定时器", name, age)  
}, 200, "why", 18);  
  
clearInterval(timerId)
```

■ 关于定时器还有一些宏任务相关的概念，我们会在JavaScript高级中讲解。

案例实战一 – 轮播消息提示



coderwhy对这件商品感兴趣

```
// 1. 获取元素
var tipBarEl = document.querySelector(".tip-bar")
var imgEl = tipBarEl.querySelector("img")
var spanEl = tipBarEl.querySelector("span")

// 2. 定时替换
var index = 0
setInterval(function() {
  var tipInfo = tipList[index]
  imgEl.src = tipInfo.icon
  spanEl.textContent = tipInfo.title
  index++
  if (index >= tipList.length) index = 0
}, 2000);
```

案例实战二 – 关闭隐藏消息



```
var topBarEl = document.querySelector(".top-bar")
var deleteEl = document.querySelector(".delete")

deleteEl.onclick = function() {
  topBarEl.style.height = 0
}

topBarEl.ontransitionend = function() {
  console.log("动画执行结束~")
  topBarEl.remove()
}
```

案例实战三 – 侧边栏展示




```
// 1. 获取元素
var iconEls = document.querySelectorAll(".icon")
var toolBarEl = document.querySelector(".tool-bar")

// 2. 动态设置
for (var i = 0; i < iconEls.length; i++) {
  var iconEl = iconEls[i]
  iconEl.style.backgroundColor = ` -48px -${i*50}px`
}

// 3. 监听鼠标事件
toolBarEl.onmouseover = function(event) {
  var iconEl = event.target
  var nameEl = iconEl.nextElementSibling
  nameEl.style.width = "62px"
}
toolBarEl.onmouseout = function(event) {
  var iconEl = event.target
  var nameEl = iconEl.nextElementSibling
  nameEl.style.width = "0"
}
```


案例实战四 – 登录框（作业）

登录页面

	邮箱/用户名/登录手机	×
---	-------------	---

	密码	 ×
---	----	--

登 录

案例实战五 – 王者荣耀tabControl

精品栏目

赛事精品

英雄攻略

精品栏目

```
var tabControlEl = document.querySelector(".tab_control")
var videoListEl = document.querySelector(".video-list")
var currentItemEl = tabControlEl.querySelector(".active")
tabControlEl.onmouseover = function(event) {
  // 1. 切换tab
  var itemEl = event.target
  if (!itemEl.classList.contains("item")) return
  itemEl.classList.add("active")
  currentItemEl.classList.remove("active")
  currentItemEl = itemEl

  // 2. 设置内容
  videoListEl.textContent = itemEl.textContent
}
```

案例实战六 – 王者轮播图

■ 见上课代码

□ 使用两种方案实现



案例实战七 – 书籍购物车

■ 见上课代码

□ 作业二：添加功能购买数量中的增量和减少购买数量；

编号	书籍名词	出版日期	价格	购买数量	操作
1	《算法导论》	2006-09	¥85	3	删除
2	《UNIX编程艺术》	2006-02	¥59	2	删除
3	《编程珠玑》	2008-10	¥39	5	删除
4	《代码大全》	2006-03	¥128	8	删除

总价格: ¥1592