# JavaScript-DOM实战案例

本文围绕JavaScript的window定时器展开，介绍了setTimeout和setInterval两种定时器方法及其使用，包括语法、参数意义和取消方法，还提及这些方法在浏览器和Node中均支持。此外，给出了轮播消息提示、登录框、王者轮播图等多个案例实战。

# 一、window定时器

## 1.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；

## 2.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

function showTime() {

console.log(new Date().toISOString());

setTimeout("showTime()", 1000);

}

showTime();

// 或者

function showTime() {

console.log(new Date().toISOString());

setTimeout(showTime, 1000);

}

showTime();

javascript

## 3.setInterval的使用

setInterval 方法和 setTimeout 的语法相同：

* 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。

*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。*

# 二、案例实战

## 1.轮播消息提示

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

<style>

.tip-bar {

display: inline-flex;

align-items: center;

height: 30px;

background-color: rgba(0,0,0,.4);

border-radius: 16px;

}

img {

width: 30px;

height: 30px;

border-radius: 50%;

margin-right: 5px;

}

span {

font-size: 13px;

color: white;

margin-right: 8px;

}

</style>

</head>

<body>

<div class="tip-bar">

<img src="https://bfs.biyao.com/group1/M01/A2/67/rBACVGA\_iOuAYaTxAAAPbted3yE165.png" alt="">

<span>183\*\*\*138对这件商品感兴趣</span>

</div>

<script>

// 1.从服务器拿到数据ajax/fetch请求

let tipList = [

{

icon: 'https://bfs.biyao.com/group1/M01/A6/97/rBACYWBCHqyAFH5tAAANZXX5Eww646.png',

title: 'coderwhy对这件商品感兴趣'

},

{

icon: 'https://bfs.biyao.com/group1/M01/A2/67/rBACVGA\_iOuAYaTxAAAPbted3yE165.png',

title: '123\*\*\*814对这件商品感兴趣'

},

{

icon: 'https://bfs.biyao.com/group1/M00/7F/4E/rBACYV16HseAP-PnAAAW9bbVoKE463.png',

title: '刘军对这件商品感兴趣'

}

]

// 2.动态的切换数据

// 2.1.获取元素

var tipBar = document.querySelector(".tip-bar")

var imgEl = tipBar.querySelector("img")

var spanEl = tipBar.querySelector("span")

// 2.2.3s切换一次数据

var currentIndex = 0 // 记录当前展示到的索引位置

setInterval(function() {

// 1> 根据索引获取item

var tipItem = tipList[currentIndex]

// 2> 给DOM设置内容

imgEl.src = tipItem.icon

spanEl.textContent = tipItem.title

// 3> 重新计算索引

currentIndex++

if (currentIndex === tipList.length) {

currentIndex = 0

}

}, 3000)

// 随机

// Math.floor(Math.random() \* tipList.length)

</script>

</body>

</html>

html

## 2.关闭隐藏信息

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

<style>

.top-bar {

display: flex;

flex-direction: row;

align-items: center;

height: 45px;

width: 375px;

background-color: black;

/\* 关键 \*/

overflow: hidden;

transition: all .5s ease-out;

}

.delete {

display: flex;

flex-direction: row;

justify-content: center;

align-items: center;

height: 100%;

width: 30px;

cursor: pointer;

}

.delete img {

height: 10px;

width: 10px;

}

.logo {

height: 30px;

width: 30px;

margin-left:3px;

margin-right: 30px;

cursor: pointer;

}

span {

color: white;

font-size: 14px;

flex: 1;

white-space: nowrap;

overflow: hidden;

text-overflow: ellipsis;

}

.btn {

width: 94px;

height: 100%;

line-height: 45px;

text-align: center;

font-size: 14px;

color: #fff;

background-color: #F63515;

}

</style>

</head>

<body>

<div class="top-bar">

<div class="delete">

<img src="./img/delete.png" alt="">

</div>

<img class="logo" src="./img/logo.png" alt="">

<span>打开京东App,购物更轻松</span>

<div class="btn">立即打开</div>

</div>

<script>

// 1.获取元素

var topBar = document.querySelector(".top-bar")

var deleteEl = topBar.querySelector(".delete")

// 2.监听delete的点击

deleteEl.onclick = function() {

topBar.style.height = 0

// setTimeout(function() {

// topBar.remove()

// }, 300)

}

// 3.监听过渡动画结束

topBar.ontransitionend = function() {

topBar.remove()

}

</script>

</body>

</html>

html

## 3.侧边栏展示

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

<style>

.tool-bar {

position: fixed;

top: 30%;

right: 0;

display: flex;

flex-direction: column;

align-items: center;

width: 35px;

}

.item {

position: relative;

width: 35px;

height: 35px;

margin-bottom: 1px;

background-color: #7a6e6e;

border-radius: 3px 0 0 3px;

}

.icon {

display: inline-block;

width: 100%;

height: 100%;

cursor: pointer;

background-image: url(./img/toolbars.png);

}

/\* .icon01 {

background-position: -48px 0;

}

.icon02 {

background-position: -48px -50px;

}

.icon03 {

background-position: -48px -100px;

}

.icon04 {

background-position: -48px -150px;

} \*/

.name {

position: absolute;

z-index: -1;

right: 35px;

/\* left: -62px; \*/

top: 0;

width: 0;

height: 35px;

line-height: 35px;

color: #fff;

text-align: center;

font-size: 12px;

background-color: #7a6e6e;

cursor: pointer;

border-radius: 3px 0 0 3px;

transition: width .2s ease;

}

.item:hover,

.item:hover .name {

background-color: #cd1926;

}

</style>

</head>

<body>

<div class="tool-bar">

<div class="item">

<i class="icon icon01"></i>

<div class="name">购物车</div>

</div>

<div class="item">

<i class="icon icon02"></i>

<div class="name">收藏</div>

</div>

<div class="item">

<i class="icon icon03"></i>

<div class="name">限时活动</div>

</div>

<div class="item">

<i class="icon icon04"></i>

<div class="name">大礼包</div>

</div>

</div>

<script>

// 1.动态给icon设置backgroundPosition

var iconEls = document.querySelectorAll(".icon")

for (var i = 0; i < iconEls.length; i++) {

var iconEl = iconEls[i]

iconEl.style.backgroundPosition = `-48px -${50\*i}px`

}

// 2.实现鼠标进入动画

// 方案一: mouseenter(不能使用事件委托)

var itemEls = document.querySelectorAll(".item")

for (var itemEl of itemEls) {

itemEl.onmouseenter = function() {

var nameEl = this.children[1]

nameEl.style.width = "62px"

}

itemEl.onmouseleave = function() {

var nameEl = this.children[1]

nameEl.style.width = "0"

}

}

// 方案二: mouseover(使用事件委托)

// var toolbarEl = document.querySelector(".tool-bar")

// toolbarEl.onmouseover = function(event) {

// handleMouseEvent(event, 62)

// }

// toolbarEl.onmouseout = function(event) {

// handleMouseEvent(event, 0)

// }

// function handleMouseEvent(event, width) {

// if (event.target !== toolbarEl) {

// // var itemEl = event.target.classList.contains("item") ? event.target: event.target.parentElement

// // 1.获取唯一的item

// var itemEl = null

// if (event.target.classList.contains("item")) {

// itemEl = event.target

// } else {

// itemEl = event.target.parentElement

// }

// // 2.根据item获取nameElement

// var nameEl = itemEl.children[1]

// // 3.设置宽度

// nameEl.style.width = `${width}px`

// }

// }

</script>

</body>

</html>

html

## 4.登录框

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

<link rel="stylesheet" href="./css/iconfont.css">

<style>

body,input {

margin: 0;

padding: 0;

outline: none;

}

.container {

text-align: center;

}

.pannel {

display: inline-block;

padding: 30px;

width: 300px;

/\* border-radius: 10px; \*/

/\* border: 2px solid #ccc; \*/

}

.pannel .form-item {

display: flex;

width: 100%;

margin-bottom: 15px;

border: 1px solid #ccc;

}

.pannel .form-item .icon {

width: 40px;

height: 40px;

box-sizing: border-box;

border-right: 1px solid #ccc;

background-color: #eee;

vertical-align: bottom;

color: #ccc;

}

.pannel .form-item .icon.icon-account {

font-size: 24px;

line-height: 40px;

}

.pannel .form-item .icon.icon-password {

line-height: 40px;

font-size: 18px;

}

.pannel .form-item input {

padding-left: 10px;

height: 40px;

flex: 1;

box-sizing: border-box;

border-style: none;

}

.pannel .form-item i {

width: 20px;

margin: 0 3px;

font-style: normal;

color: #ccc;

line-height: 40px;

font-size: 20px;

}

button {

width: 100%;

background-color: rgb(233, 97, 99);

border-style: none;

padding: 5px 0;

font-size: 20px;

letter-spacing: 5px;

color: #fff;

}

</style>

</head>

<body>

<div class="container">

<div class="pannel">

<form action="/abc" method="post">

<h1>登录页面</h1>

<div class="form-item">

<div class="icon iconfont icon-account"></div>

<input type="text" name="username" id="username" placeholder="邮箱/用户名/登录手机">

<i class="remove-icon iconfont icon-delete"></i>

</div>

<div class="form-item">

<div class="icon iconfont icon-password"></div>

<input type="password" name="password" id="password" placeholder="密码">

<i class="hidden-icon iconfont icon-no\_eye"></i>

<i class="remove-icon iconfont icon-delete"></i>

</div>

<button type="submit">登录</button>

</form>

</div>

</div>

<script>

var removeIconEls = document.querySelectorAll(".icon-delete");

removeIconEls.forEach(function (value) {

console.log(value);

value.addEventListener("click", function () {

console.log("当前元素：", this);

this.previousElementSibling.value = "";

})

})

</script>

</body>

</html>

html

## 5.王者荣耀tabControl

<!DOCTYPE html>

<html lang="zh-CN">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>王者荣耀-main-news</title>

<link rel="stylesheet" href="./css/reset.css">

<link rel="stylesheet" href="./css/common.css">

<style>

.main .section-content {

display: flex;

justify-content: space-between;

}

.main .section-content .left-content {

width: 872px;

height: 1000px;

}

.main .section-content .right-content {

width: 295px;

height: 500px;

}

</style>

</head>

<body>

<div class="main main\_wrapper">

<div class="section-content">

<div class="left-content">

<div class="content-center">

<div class="section\_header">

<div class="header\_left">

<h3 class="title">内容中心</h3>

</div>

<div class="header\_right" href="#">

<a class="more" href="#">更多</a>

</div>

</div>

<div class="tab\_control">

<div class="item active">精品栏目</div>

<div class="line"></div>

<div class="item">赛事精品</div>

<div class="line"></div>

<div class="item">英雄攻略</div>

</div>

</div>

</div>

</div>

<script>

// 1.获取元素

var tabControl = document.querySelector(".tab\_control")

// 2.监听鼠标进入(事件委托)

var activeLiEl = tabControl.querySelector(".active")

tabControl.onmouseover = function(event) {

// 1.拿到事件发生的对象

var itemEl = event.target

if (itemEl.classList.contains("item")) {

// 其他的取消active

// 1.for循环所有的item

// 2.querySelector(".active")

// 3.记录当前的active对应的item

activeLiEl.classList.remove("active")

// 当前进入的item变成active

itemEl.classList.add("active")

// 将最新的itemEl变成activeLiEl

activeLiEl = itemEl

}

}

</script>

</body>

</html>

html

## 6.王者轮播图

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

<style>

body {

margin: 0;

padding: 0;

}

a {

text-decoration: none;

}

.box {

width: 604px;

height: 342px;

margin: 0 auto;

overflow: hidden;

}

.pic {

display: flex;

width: 100%;

height: 298px;

transition: all 300ms ease;

}

.pic .banner {

flex: 1;

}

.btn {

display: flex;

height: 44px;

background-color: #000;

}

.btn a {

flex: 1;

line-height: 44px;

font-size: 14px;

text-align: center;

color: #b1b2be;

}

.btn a.active,

.btn a:hover {

color: #f3c258;

background-color: rgba(255, 255, 255, 0.15);

}

</style>

</head>

<body>

<div class="box">

<div class="pic">

<a href="#" class="banner"><img src="./images/banner01.jpeg" alt="海诺设计理念"></a>

<a href="#" class="banner"><img src="./images/banner02.jpeg" alt="季后赛宣传片"></a>

<a href="#" class="banner"><img src="./images/banner03.jpeg" alt="KPL选秀招募"></a>

<a href="#" class="banner"><img src="./images/banner04.jpeg" alt="荣耀大话王"></a>

<a href="#" class="banner"><img src="./images/banner05.jpeg" alt="虔诚人物志"></a>

</div>

<div class="btn">

<a href="#" class="active">海诺设计理念</a>

<a href="#">季后赛宣传片</a>

<a href="#">KPL选秀招募</a>

<a href="#">荣耀大话王</a>

<a href="#">虔诚人物志</a>

</div>

</div>

<script>

// 获取banner

var boxEl = document.querySelector(".box");

var picEl = document.querySelector(".pic");

var btnEl = document.querySelector(".btn");

var index = 0;

var activeBtn = document.querySelector(".active");

// 轮播图

var bannerTimer = exeBanner();

// 按钮鼠标切换

boxEl.onmouseover = function (event) {

// 暂停轮播

clearInterval(bannerTimer);

// 判断是否为按钮，不是则返回

if (event.target.parentElement !== btnEl ||

event.target === activeBtn) return;

//切换按钮激活

changeActive(event.target);

// 切换对应的banner

// 找到当前按钮的index

index = Array.from(btnEl.children).indexOf(activeBtn);

picEl.style.transform = `translate(-${604\*index}px, 0)`;

}

boxEl.onmouseleave = function (event) {

// 开始轮播

bannerTimer = exeBanner();

}

// 改变激活按钮元素

function changeActive(activeEl) {

activeBtn.classList.remove("active");

activeBtn = activeEl;

activeBtn.classList.add("active");

}

// 执行轮播

function exeBanner(params) {

return setInterval(() => {

// banner图移动

picEl.style.transform = `translate(-${604\*index}px, 0)`;

// 按钮切换

changeActive(btnEl.children[index]);

index++;

if (index === picEl.children.length) {

index = 0;

}

}, 3000);

}

</script>

</body>

</html>

html

### 原版

<!DOCTYPE html>

<html lang="zh">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

<style>

body {

margin: 0;

padding: 0;

}

a {

text-decoration: none;

}

.box {

width: 604px;

height: 342px;

margin: 0 auto;

overflow: hidden;

}

.pic {

display: flex;

position: relative;

width: 100%;

height: 298px;

}

.pic .banner {

position: absolute;

flex: 1;

/\* transition: all 2000ms ease; \*/

}

.btn {

display: flex;

height: 44px;

background-color: #000;

}

.btn a {

flex: 1;

line-height: 44px;

font-size: 14px;

text-align: center;

color: #b1b2be;

}

.btn a.active,

.btn a:hover {

color: #f3c258;

background-color: rgba(255, 255, 255, 0.15);

}

</style>

</head>

<body>

<div class="box">

<div class="pic">

<a href="#" class="banner"><img src="./images/banner01.jpeg" alt="海诺设计理念"></a>

<a href="#" class="banner"><img src="./images/banner02.jpeg" alt="季后赛宣传片"></a>

<a href="#" class="banner"><img src="./images/banner03.jpeg" alt="KPL选秀招募"></a>

<a href="#" class="banner"><img src="./images/banner04.jpeg" alt="荣耀大话王"></a>

<a href="#" class="banner"><img src="./images/banner05.jpeg" alt="虔诚人物志"></a>

</div>

<div class="btn">

<a href="#" class="active">海诺设计理念</a>

<a href="#">季后赛宣传片</a>

<a href="#">KPL选秀招募</a>

<a href="#">荣耀大话王</a>

<a href="#">虔诚人物志</a>

</div>

</div>

<script>

// 获取banner

var boxEl = document.querySelector(".box");

var picEl = document.querySelector(".pic");

var btnEl = document.querySelector(".btn");

var index = 0;

var lastIndex = index;

var activeBtn = document.querySelector(".active");

// 轮播图

var bannerTimer = exeBanner();

// 按钮鼠标切换

boxEl.onmouseover = function (event) {

// 暂停轮播

clearInterval(bannerTimer);

// 判断是否为按钮，不是则返回

if (event.target.parentElement !== btnEl ||

event.target === activeBtn) return;

//切换按钮激活

changeActive(event.target);

// 切换对应的banner

// 找到当前按钮的index

lastIndex = index;

index = Array.from(btnEl.children).indexOf(activeBtn);

bannerMove();;

}

boxEl.onmouseleave = function (event) {

// 开始轮播

bannerTimer = exeBanner();

}

// 改变激活按钮元素

function changeActive(activeEl) {

activeBtn.classList.remove("active");

activeBtn = activeEl;

activeBtn.classList.add("active");

}

// 执行轮播

function exeBanner(params) {

return setInterval(() => {

// 切换轮播图

bannerMove();

// 切换按钮

changeActive(btnEl.children[index]);

lastIndex = index;

index++;

// 重置index

if (index === picEl.children.length) {

index = 0;

}

}, 3000);

}

// 轮播图移动

function bannerMove() {

// 给上一张和下一张添加动画

for (var i = 0; i < picEl.children.length; i++) {

if (i === lastIndex || i === index) {

picEl.children[i].style.transition = "all 300ms ease";

continue;

}

picEl.children[i].style.transition = "none";

}

// banner图移动

if (index > 0) {

for (var i = 0; i < index; i++) {

picEl.children[i].style.left = "-100%";

}

}

if (index === 0) {

for (var i = 1; i < picEl.children.length; i++) {

picEl.children[i].style.left = "100%";

}

}

picEl.children[index].style.left = "0%";

if (index < picEl.children.length - 1) {

for (var i = index+1; i < picEl.children.length; i++) {

picEl.children[i].style.left = "100%";

}

}

}

</script>

</body>

</html>

html

## 7.书籍购物车

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

<style>

table {

border-collapse: collapse;

}

thead {

background-color: #f5f5f5;

}

th, td {

border: 1px solid #aaa;

padding: 8px 12px;

text-align: center;

}

</style>

</head>

<body>

<table>

<thead>

<tr>

<th>编号</th>

<th>书籍名称</th>

<th>出版日期</th>

<th>价格</th>

<th>购买数量</th>

<th>操作</th>

</tr>

</thead>

<tbody>

</tbody>

</table>

<h2 class="price">

总价格: ¥<span class="price-count">0</span>

</h2>

<script>

// 1.从服务器获取数据 ajax/fetch

var books = [

{

id: 1,

name: '《算法导论》',

date: '2006-09',

price: 85.00,

count: 3

},

{

id: 2,

name: '《UNIX编程艺术》',

date: '2006-02',

price: 59.00,

count: 2

},

{

id: 3,

name: '《编程珠玑》',

date: '2008-10',

price: 39.00,

count: 5

},

{

id: 4,

name: '《代码大全》',

date: '2006-03',

price: 128.00,

count: 8

}

]

// 2.对数据展示

// 到底通过html直接编写, 还是通过JavaScriptDOM操作创建元素

// 1> 对于固定的, 直接通过html编写(能通过html编写, 尽量通过html直接编写)

// 2> 对于哪些大量的数据, 有规律的数据, 可以通过JavaScript编写

var tbodyEl = document.querySelector("tbody")

// 2.2. 动态添加tr以及内部数据

for (var i = 0; i < books.length; i++) {

var trowEl = document.createElement("tr")

// 2.3. 放具体数据

var book = books[i]

var bookKeys = Object.keys(book)

for (var m = 0; m < bookKeys.length; m++) {

var key = bookKeys[m]

var value = book[key]

var tdEl = document.createElement("td")

if (key === "price") {

value = "¥" + value

}

tdEl.textContent = value

trowEl.append(tdEl)

}

// 2.4. 添加删除按钮

var deleteTdEl = document.createElement("td")

var deleteBtnEl = document.createElement("button")

deleteBtnEl.textContent = "删除"

deleteTdEl.append(deleteBtnEl)

trowEl.append(deleteTdEl)

// 2.5.监听删除按钮的点击

deleteBtnEl.onclick = function() {

// 1.删除对应的trow

var deleteTRowEl = this.parentElement.parentElement

var deleteTrIndex = deleteTRowEl.sectionRowIndex

deleteTRowEl.remove()

// 2.删除对应books中的数据

books.splice(deleteTrIndex, 1)

// 3.重新计算一次价格

calcTotalPrice()

}

tbodyEl.append(trowEl)

}

// 3.计算总价格

var priceCountEl = document.querySelector(".price-count")

calcTotalPrice()

// 封装计算价格的函数

function calcTotalPrice() {

var totalPrice = books.reduce(function(preValue, item) {

return preValue + item.count \* item.price

}, 0)

priceCountEl.textContent = totalPrice

}

</script>

</body>

</html>

html