

Web APIs 第五天

DOM- 网页特效篇



目录

Contents

- ◆ 滚动事件和加载事件
- ◆ 元素大小和位置
- ◆ 综合案例

学习目标

Learning Objectives

1. 能够制作常见的网页特效
2. 具备制作常见网页焦点图的能力



滚动事件和加载事件

- 滚动事件
- 加载事件

目标：掌握新的事件，做更强交互

一、滚动事件和加载事件

1.1 滚动事件

- 当页面进行滚动时触发的事件
- 为什么要学？
 - 很多网页需要检测用户把页面滚动到某个区域后做一些处理，比如固定导航栏，比如返回顶部
- 事件名：scroll
- 监听整个页面滚动：

```
// 页面滚动事件
window.addEventListener('scroll', function () {
    // 执行的操作
})
```

```
} } ➢ 给 window 或 document 添加 scroll 事件
```

- 监听某个元素的内部滚动直接给某个元素加即可



滚动事件和加载事件

- 滚动事件
- 加载事件

一、滚动事件和加载事件

1.2 加载事件

- 加载外部资源（如图片、外联CSS和JavaScript等）加载完毕时触发的事件
- 为什么要学？
 - 有些时候需要等页面资源全部处理完了做一些事情
 - 老代码喜欢把 script 写在 head 中，这时候直接找 dom 元素找不到
- 事件名：load
- 监听页面所有资源加载完毕：
 - 给 window 添加 load 事件

```
// 页面加载事件
window.addEventListener('load', function () {
    // 执行的操作
})
})
```

- 注意：不光可以监听整个页面资源加载完毕，也可以针对某个资源绑定load事件

一、滚动事件和加载事件

1.2 加载事件

- 当初始的 HTML 文档被完全加载和解析完成之后，DOMContentLoaded 事件被触发，而无需等待样式表、图像等完全加载
- 事件名：DOMContentLoaded
- 监听页面DOM加载完毕：
 - 给 document 添加 DOMContentLoaded 事件

```
document.addEventListener('DOMContentLoaded', function () {  
    // 执行的操作  
})
```




总结

1. 页面滚动事件如何添加？

- scroll
- 监听整个页面滚动给 window 或 document 加

2. 加载事件有哪两个？如何添加？

- load 事件
 - 监听整个页面资源给 window 加
- DOMContentLoaded
 - 给 **document** 加,当初始的 HTML 文档被完全加载和解析完成之后, DOMContentLoaded 事件被触发, 而无需等待样式表、图像等完全加载



目录

Contents

- ◆ 滚动事件和加载事件
- ◆ 元素大小和位置
- ◆ 综合案例



元素大小和位置

- scroll家族
- offset家族
- client家族

- 目标：掌握元素大小和位置的获取方法，为后续网页特效打基础

2.1 scroll家族

- 使用场景：

我们想要页面滚动一段距离，比如100px，就让某些元素

显示隐藏，那我们怎么知道，页面滚动了100像素呢？

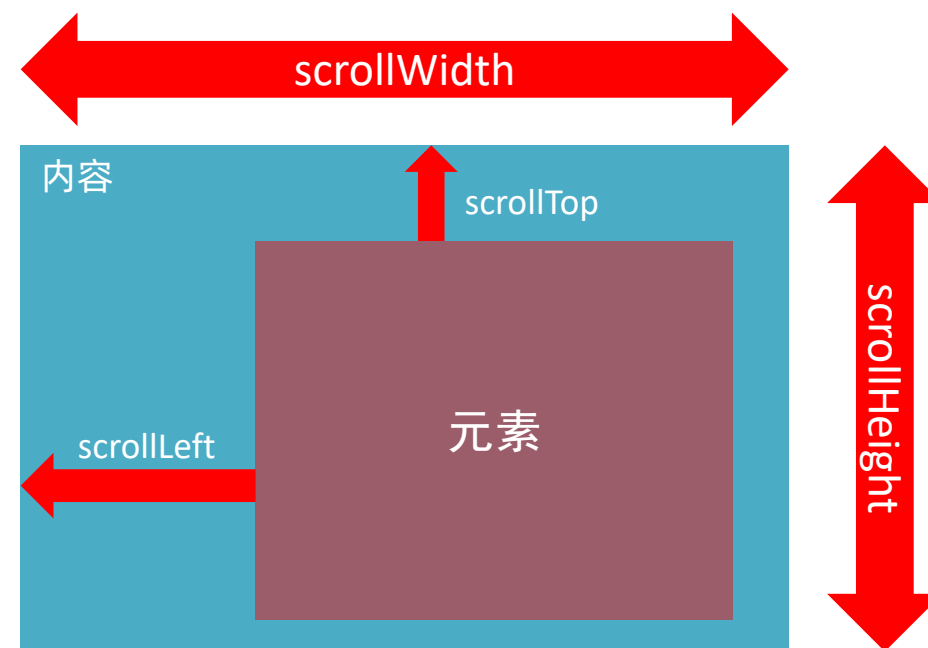
就可以使用scroll 来检测页面滚动的距离~~~

二、元素大小和位置

2.1 scroll家族

- 获取宽高：
 - 获取元素的**内容**总宽高（不包含滚动条）返回值不带单位
 - scrollWidth和scrollHeight
- 获取位置：
 - 获取元素内容往左、往上滚出去看不到的距离
 - **scrollLeft**和**scrollTop**
 - 这两个属性是可以**修改**的

```
div.addEventListener('scroll', function () {  
    console.log(this.scrollTop)  
})
```



2.1 scroll家族

- 开发中，我们经常检测页面滚动的距离，比如页面滚动100像素，就可以显示一个元素，或者固定一个元素

```
// 页面滚动事件
window.addEventListener('scroll', function () {
    // document.documentElement.scrollTop 获得当前页面被卷去的头部
    let num = document.documentElement.scrollTop
    console.log(num)
})
```

注意事项

document.documentElement HTML 文档返回对象为HTML元素

```
<html lang="en">
  <head>...</head>
  <body>...</body>
</html>
```

总结

1. scrollWidth和scrollHeight是得到元素什么的宽高?
 - 内容
 - 不包含滚动条
2. 被卷去的头部或者左侧用那个属性？是否可以读取和修改？
 - scrollTop / scrollLeft
 - 可以读取，也可以修改（赋值）
3. 检测页面滚动的头部距离（被卷去的头部）用那个属性？
 - document.documentElement.scrollTop

```
// 页面滚动事件
window.addEventListener('scroll', function () {
    // document.documentElement.scrollTop 获得当前页面被卷去的头部
    let num = document.documentElement.scrollTop
    console.log(num)
})
```

案例

页面滚动显示返回顶部按钮

需求：当页面滚动500像素，就显示返回顶部按钮，否则隐藏，同时点击按钮，则返回顶部

分析：

- ①：用到页面滚动事件
- ②：检测页面滚动大于等于100像素，则显示按钮
- ③：点击按钮，则让页面的scrollTop 重置为 0





元素大小和位置

- scroll家族
- offset家族
- client家族

- 目标：掌握元素大小和位置的获取方法，为后续网页特效打基础

二、元素大小和位置

2.2 offset家族

- 使用场景：

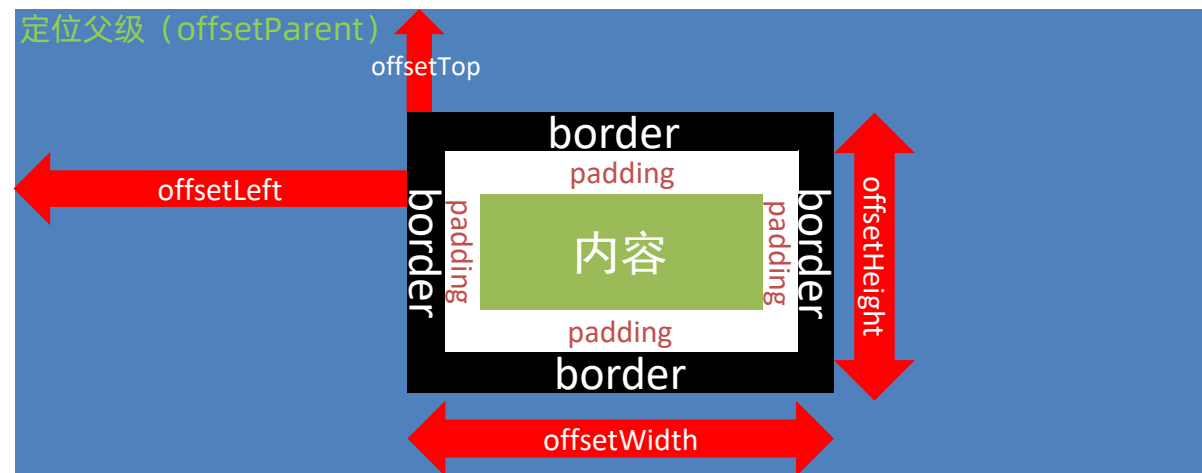
前面案例滚动多少距离，都是我们自己算的，最好是页面滚动到某个元素，就可以做某些事。

简单说，就是通过js的方式，得到元素在页面中的位置
这样我们可以做，页面滚动到这个位置，就可以返回顶部的小盒子显示...



2.2 offset家族

- 获取宽高：
 - 获取元素的自身宽高、包含元素自身设置的宽高、padding、border
 - offsetWidth和offsetHeight
- 获取位置：
 - 获取元素距离自己**定位父级**元素的左、上距离
 - **offsetLeft和offsetTop** 注意是只读属性





总结

1. `offsetWidth`和`offsetHeight`是得到元素什么的宽高?
 - 内容 + padding + border
2. `offsetTop`和`offsetLeft` 得到位置以谁为准?
 - 带有定位的父级
 - 如果都没有则以 文档左上角 为准

案例

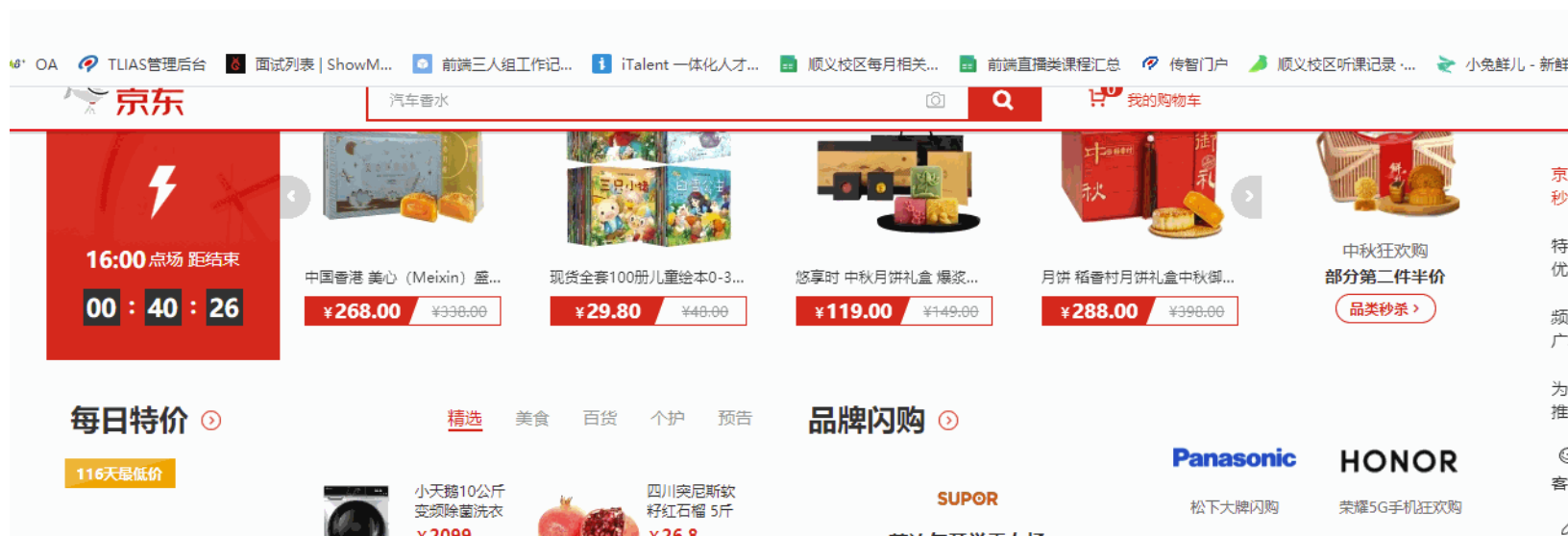
仿京东固定导航栏案例

需求：当页面滚动到秒杀模块，导航栏自动滑入，否则滑出

分析：

①：用到页面滚动事件

②：检测页面滚动大于等于 秒杀模块的位置 则滑入，否则滑出



案例

电梯导航案例

需求：点击可以页面调到指定效果

分析：

- ①：点击当前 小导航，当前添加active，其余移除active
- ②：得到对应 内容 的 offsetTop值
- ③：让页面的 scrollTop 走到 对应 内容 的 offsetTop



元素大小和位置

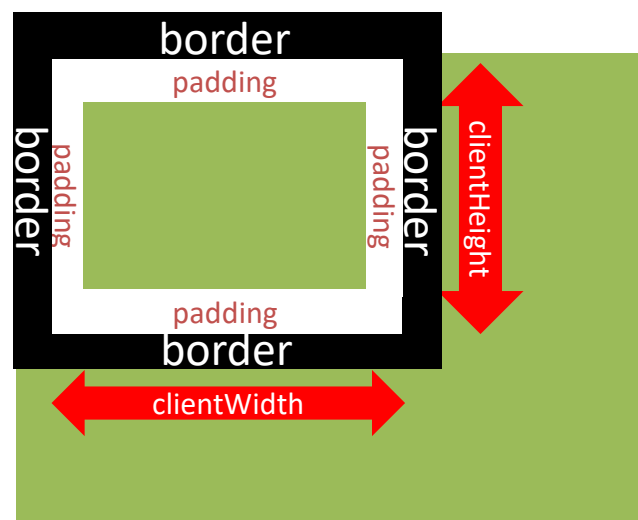
- scroll家族
- offset家族
- client家族

- 目标：掌握元素大小和位置的获取方法，为后续网页特效打基础

二、元素大小和位置

2.3 client家族

- 获取宽高：
 - 获取元素的可见部分宽高（不包含边框，滚动条等）
 - clientWidth和clientHeight
- 获取位置：
 - 获取左边框和上边框宽度
 - clientLeft和clientTop 注意是只读属性



二、元素大小和位置

2.3 client家族

- 会在窗口尺寸改变的时候触发事件：

➤ resize

```
window.addEventListener('resize', function () {  
    // 执行的代码  
})
```

- 检测屏幕宽度：

```
window.addEventListener('resize', function () {  
    let w = document.documentElement.clientWidth  
    console.log(w)  
})
```



总结

1. offset家族

- 获取元素自身大小：包括自身设置的宽高、padding、border
- 获取元素距离定位父级的左和上距离 只读属性

2. client家族

- 获取元素可见区域的大小
- 获取元素左、上边框距离 只读属性

3. scroll家族

- 获取元素内容的总大小
- 获取元素向左向上滚出去看不见的距离 可读写属性



目录

Contents

- ◆ 滚动事件和加载事件
- ◆ 元素大小和位置
- ◆ 综合案例

案例

轮播图案例

需求：



案例

轮播图案例

分析：

需求①：小图标鼠标经过事件

鼠标经过小图片，当前高亮，其余兄弟变淡 添加类

需求②：大图片跟随变化

对应的大图片跟着显示，如果想要过渡效果，可以使用opacity效果，可以利用CSS淡入淡出的效果，还是添加类

需求③：右侧按钮播放效果

点击右侧按钮，可以自动播放下一张图片

需要一个变化量 index 不断自增

然后播放下一张图片

如果到了最后一张，必须要还原为第1张图片

教你一招：索引号 = 索引号 % 数组长度（放到播放前面）

案例

轮播图案例

分析：

需求④：解决一个BUG

点击右侧按钮可以实现播放下一张，但是鼠标经过前面的，播放就会乱序

解决方案： 让变化量（索引号）重新赋值为 当前鼠标经过的索引号

需求⑤：左侧按钮播放效果

点击左侧按钮，可以自动播放上一张图片

需要一个变化量 `index` 不断自减

然后播放上一张图片

如果到了第一张，必须要从最后一张播放

教你一招： $\text{索引号} = (\text{数组长度} + \text{索引号}) \% \text{数组长度}$

需求⑥：

因为左侧按钮和右侧按钮里面有大量相同的操作，可以抽取封装一个函数 `common`

案例

轮播图案例

分析：

需求⑦：开启定时器

其实定时器自动播放，就相当于点击了右侧按钮，此时只需要，`right.click()`

需求⑧：

鼠标经过停止定时器（清除定时器）

鼠标离开开启定时器（开启定时器）



思考

1. 整理今天笔记
2. 晚上再写2遍轮播图案例

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