昨日知识点回顾：

01、移动web开发的基础知识

02、远成调试

03、视口

04、屏幕适配

05、京东首页的切图

今日知识点：

01、京东页面产品区域的完成

02、京东顶部搜索和倒计时效果

03、轮播图样式结构修改

04、移动端touch事件 (重点)

05、轮播图实现滑动效果

# 1 京东页面产品区域的完成

## 1.1 01-移动web-产品块制作(图片区域上)

结构分析：

1. 对于秒杀和京东超市有公共区域就用公共的样式编写
2. 伪元素 :before和 :after 的使用，containt:””，以及绝对定位不能忘了

3、京东超市区块：左边和右边布局 添加左右浮动即可，**切记：一定要清除浮动**

## 1.2 02-移动web-产品块制作(图片区域下)

实现图片下面阴影运用的是CSS3中所学的

box-shadow ：1px 1px 1px 1px #ccc

## 1.3 03-移动web-产品块制作(秒杀标题栏)

**结构分析：**

1. 头部：左边左浮动，右边右浮动
2. 掌上秒杀和产品块用了一个公共的头部样式，为了只更改掌上秒杀需要重新定义一个样式，也就是 .js\_sk{}
3. 选择器 nth-of-type(3n) / nth-last-of-type(3n) 找到三的倍数的元素

nth-of-type(3n)：正数

nth-last-of-type(3n) 倒数

## 1.4 04-移动web-产品块制作(秒杀块内容)

**结构步骤：**

1. 最外层盒子类名为 .jd\_pContent 设置内边距 padding
2. 每个图片盒子li 设置浮动均分成三等分、内容居中显示
3. 图片可以点击用a标签包裹课链接，图片需要设置尺寸
4. 最后一定不能忘了清除浮动

# 2、京东顶部搜索和倒计时效果

## 2.1 05-移动web-京东首页搜索块js效果

1. 给头部设置背景颜色rgba() 透明值为0
2. 分析渐变效果
   1. 开始为滚动透明度为0，随着滚动的距离加大，背景颜色逐渐加深
   2. 滚动超过一定高度时，背景颜色就不在改变
3. js效果实现步骤
   1. 引入的是外部js需要 js页面加载完入口函数: window.onload=function(){}
   2. 用CSS3的选择器quertSelector() 获去轮播图和搜索盒子对象，并得到轮播图盒子的高度
   3. 滚动事件用到 window.onscroll =function(){}，同时对滚动距离和轮播图高度进行判断 从而得到opacity的值

window.onscroll=function(){  
 var offsetTop=document.body.scrollTop;  
 /\*3.计算比例值，获取透明度，设置背景颜色的样式\*/  
 var opacity=0;  
 /\*判断，如果当banner还没有完全 滚出屏幕，那么才有必要计算透明度和设置透明度\*/  
 if(offsetTop < bannerHeight){  
 opacity=offsetTop/bannerHeight;  
 /\*设置样式\*/  
search.style.backgroundColor="rgba(233,35,34,"+opacity+")";  
 }  
}

## 2.2 06-移动web-京东首页倒计时js效果

a/b 这个是取整 求商

a%b 这个是 求余

**Math**.floor()向下取整

**一、步骤分析：**

1. 获取所有的展示时间的span
2. 设置需要定时的时间
3. 开启定时器 (一秒一次自减 )
4. 计算时分秒

/\*获取时\*/   
var hour=*Math*.floor(totalTime/3600);   
/\*获取分\*/  
var minute=*Math*.floor(totalTime%3600/60);   
/\*获取秒\*/   
var second=*Math*.floor(totalTime%60);   
/\*赋值，将时间填充到span中 12\*/   
spans[0].innerHTML=*Math*.floor(hour/10);  
spans[1].innerHTML=*Math*.floor(hour%10);

spans[3].innerHTML=*Math*.floor(minute/10);  
spans[4].innerHTML=*Math*.floor(minute%10);

spans[6].innerHTML=*Math*.floor(second/10);  
spans[7].innerHTML=*Math*.floor(second%10);

1. 判断定时的时间小于0就清除定时器

if(totalTime < 0){  
 /\*清除时钟\*/  
 clearInterval(timerId);  
 return;  
}

# 3 轮播图样式结构修改

## 3.1 07-移动web-京东首页-轮播图(拖动修改页面结构和样式)

**思路：**

1.在首尾添加图片

a).在开始位置添加原始的最后一张图片

b).在最后位置添加原始的第一张图片

2.修改页面结构

3.修改对应的样式

.jd\_bannerImg{

width:1000%;

}

.jd\_bannerImg > li{

width:10%;

float: left;

}

4.设置默认的偏移：默认应该显示索引1的图片

transform:translateX(-10%):参照自身

left:-100%:参照父容器

使用定位实现的细节：当前元素必须使用**relative**定位，否则父容器无法获取正确的高度

## 3.2 08-移动web-京东首页-轮播图(js添加首尾图片)

1. **选择器的回顾**
   1. first-of-type
   2. last-of-type
2. **深度克隆+添加元素+插入位置** 
   1. cloneNode(true)
   2. appendChild()
   3. insertBefore(插入对象，位置)

/\*1.1.获取轮播图结构\*/  
var banner=document.querySelector(".jd\_banner");  
/\*1.2.获取图片容器\*/  
var imgBox=banner.querySelector("ul:first-of-type");  
/\*1.3.获取原始的第一张图片\*/  
var first=imgBox.querySelector("li:first-of-type");  
/\*1.4.获取原始的最后一张图片\*/  
var last=imgBox.querySelector("li:last-of-type");  
/\*1.5.在首尾插入两张图片 cloneNode:复制一个dom元素\*/  
imgBox.appendChild(first.cloneNode(true));  
/\*insertBefore(需要插入的dom元素，位置)\*/  
imgBox.insertBefore(last.cloneNode(true),imgBox.firstChild);

## 3.3 09-移动web-京东首页-轮播图(js设置样式)

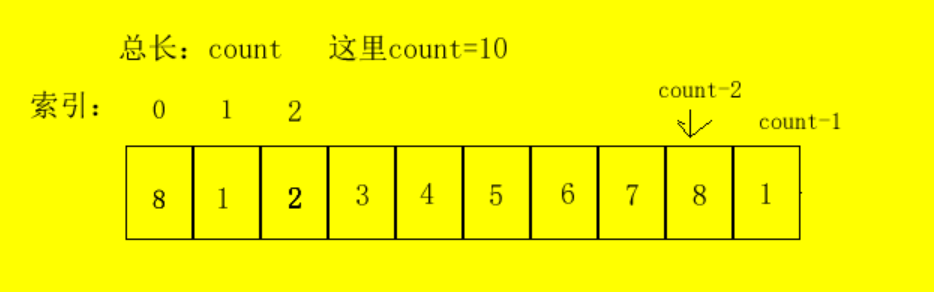
**步骤：**

**1、通过js设置整个轮播图盒子的宽度以及定位**

* 1. 获取所有的li的个数
  2. 获取轮播图的宽度
  3. 整个轮播图盒子的宽度=所有的li的个数\*轮播图的宽度
  4. 设置每个li的宽度，通过遍历获取每个li，在给其赋值为轮播图的宽度
  5. 设置第二张图片首先显示，需要让其位偏移一个负值的轮播图的宽度

/\*2.设置对应的样式\*/  
/\*2.1获取所有li元素\*/  
var lis=imgBox.querySelectorAll("li");  
/\*2.2 获取li元素的数量\*/  
var count=lis.length;  
/\*2.3.获取banner的宽度\*/  
var bannerWidth=banner.offsetWidth;  
/\*2.4 设置图片盒子的宽度\*/  
imgBox.style.width=count\*bannerWidth+"px";  
/\*2.5 设置每一个li(图片)元素的宽度\*/  
for(var i=0;i<lis.length;i++){  
 lis[i].style.width=bannerWidth+"px";  
};  
/\*3.设置默认的偏移\*/  
imgBox.style.left=-bannerWidth+"px";

1. **屏幕改变时重新计算宽度**
   1. 使用onresize事件对屏幕实时监听
   2. 在onresize事件监听下重新再获取轮播图的宽度+图片盒子的宽度+每个li的宽度+图片盒子的位偏移

/\*4.当屏幕变化的时候，重新计算宽度\*/  
window.onresize=function(){  
 /\*4.1.获取banner的宽度,覆盖全局的宽度值\*/  
 bannerWidth=banner.offsetWidth;  
 /\*4.2 设置图片盒子的宽度\*/  
 imgBox.style.width=count\*bannerWidth+"px";  
 /\*4.3设置每一个li(图片)元素的宽度\*/  
 for(var i=0;i<lis.length;i++){  
 lis[i].style.width=bannerWidth+"px";  
 };  
 /\*4.4重新设置定位值\*/  
 imgBox.style.left=-index\*bannerWidth+"px";  
};

## 3.4 10-移动web-京东首页-轮播图(js实现自动轮播)

**1、自动轮播**

* 1. setInterval(function(),2000) 定时器2s一次
  2. setTimeout(function(){},500) 定时器只执行一次，有500毫秒的延迟

1. **自动轮播+无缝轮播的原理**
   1. 自动轮播原理
      1. 首先申明全局的index=1(因位偏移首先显示的index=1的那张图片)
      2. 定时器1s执行一次，让索引自增一次，
      3. 设置图片盒子的位偏移 =-当前的索引\*图片盒子的宽度
      4. 添加过渡让其有动画效果

imgBox.style.transition="left 0.5s ease-in-out";  
imgBox.style.left=(-index\*bannerWidth)+"px";

b) 无缝轮播的原理

i. 添加定时延长

ii. 判断索引值为9(也就是最后一张),直接让其索引index=1

iii. 设置图片盒子的位偏移 = -当前的索引\*图片盒子的宽度

Iv. 关闭过渡

setTimeout(function(){  
 if(index==count-1){  
 console.log(index);  
 index=1;  
 imgBox.style.transition="none";  
 imgBox.style.left=(-index\*bannerWidth)+"px";  
 }  
},500);

## 3.5 11-移动web-京东首页轮播图自动轮播完善

**1、为什么会出现屏幕变化时图片自动轮播还是会有错乱效果**

onresize事件监听后重新定义了一个bannerWidth ，而轮播图在定时器的作用下对图片盒子进行的位偏移值(-index\*bannerWidth)这里的bannerWidth还是调用的全局变量那个值，并不是获取屏幕改变后的值，为解决这个问题，我们只需要将监听的值覆盖全局变量的值即可

# 4 移动端touch事件

## 4.1 12-移动web-移动端的touch事件简介

**1、touch事件有几种及什么情况下触发？**

* 1. touchsatrt 当手指触摸屏幕的时候触发
  2. touchmove 当手指在屏幕上滑动的时候连续触发
  3. touchend 当手指离开的时候触发

**2、touch事件触发的顺序**

touchsatrt > touchmove > touchend

## 4.2 13-移动web-移动端的touch事件(手指对象数组) 1、touches、targetTouches、changedTouches的意思是什么？

1. changedTouches 当前屏幕上变换的手指对象 从无到有或从有到无 (touchsatrt时包含与触摸屏接触的触点，touchend时包含离开触摸屏的触点)
2. targetTouches 位于该元素上的所有手指的列表
3. touches 位于屏幕上的所有手指列表

注：没有对比出touches同targetTouches的差异，推荐使用targetTouches

## 4.3 14-移动web-移动端的touch事件(手指对象的坐标获取)

**1、touches下面获取坐标是哪几个属性均是什么作用？**

* 1. screenX /screenY:手指的触摸点相对于屏幕左上角的坐标距离
  2. clientX /clientY:相对于当前窗口--移动端屏幕
  3. pageX /pageY:相对于当前页面的内容 会有滚动条 包含滚动条的

## 4.4 15-移动web-使用touch事件实现元素的拖拽

1. **元素实现拖拽的原理是什么？**
   1. 获取当前触摸点的坐标
   2. 获取滑动点的坐标
   3. 获取两者的差值
   4. 把差值传给对象就是它的位移值了

div.addEventListener("touchstart",function(*e*){  
 *startX*= *e*.targetTouches[0].clientX;  
 *startY*= *e*.targetTouches[0].clientY;  
 console.log(*startX*+":"+*startY*);  
});  
/\*触摸滑动:持续触发\*/  
div.addEventListener("touchmove",function(*e*){  
 /\*记录手指滑动过程中的坐标值\*/  
 *moveX*= *e*.targetTouches[0].clientX;  
 *moveY*= *e*.targetTouches[0].clientY;  
 console.log(*moveX*+":"+*moveY*);  
 /\*计算与上一次坐标的差异\*/  
 *distanceX*=*moveX*-*startX*;  
 *distanceY*=*moveY*-*startY*;  
 console.log(*distanceX*+":"+*distanceY*);  
 /\*设置偏移\*/  
 div.style.transform="translate("+*distanceX*+"px,"+*distanceY*+"px)";  
});

## 4.5 16-移动web-touch事件源参数中的target

1. **任意对象能拖动原理**

通过对document添加事件可以捕获当前响应事件的对象,然后再获取当前对象的坐标来实现拖动偏移定位

1. **如何使用target**

通过对document添加事件可以捕获当前响应事件的对象e，从而获取当前响应事件对象e的target

# 5 轮播图实现滑动效果

## 5.1 17-移动web-轮播图-实现滑动操作

1. **touch事件的触发有什么前提条件**

**必须保证元素有具体的宽高，如果宽或高为0，则不会进行触发**

**2、分析实现手动轮播方法**

1.手指触发轮播图时清除定时器

2.记录手指的起始位置

3.记录手指在滑动过程中的位置，计算出相对于起始位置的偏移值，通过left样式实现图片的偏移 (最后这个偏移是当前图片相对于起始位置的偏移值+当前滑动的距离)

imgBox.style.left=(-index\*bannerWidth + distanceX)+"px";

4.在松开手指之后，判断当前滑动的距离，如果超出指定的范围，就翻页，否则回弹

5.松开手指之后，重新开启定时器

## 5.2 18-移动web-轮播图-实现滑动操作(细节)

1. **滑动时图片产生延迟的原因是什么**

开启定时器的时候添加了过渡效果，在touchmove的时候没有清除这个过渡也就导致那个延迟了，解决的办法是在touchmove事件中清除过渡

## 5.3 19-移动web-轮播图-松开手指之后的处理

1. **滑动的方向如何确定**
   1. 根据distance从左往右为正值 (index--)

b) 根据distance从右往左为负值 (index++)

1. **松开手指后如何实现上一张或者下一张切换**

**原理分析：**

**A:滑动大于100时**

1. .通过滑动的距离(distance)的绝对值大于100时再来判断其方向
2. .滑动的方向根据distance值为正还是为负
3. .为负：从右往左滑 index++ ,为正：从左往右滑 index--
4. .给图片盒子添加过渡效果
5. .图片盒子位偏移=-当前的索引\*图片盒子的宽度

**B:有滑动且滑动距离大于0**

1. .给图片盒子添加过渡效果

b).图片盒子位偏移=-当前的索引\*图片盒子的宽度

**C:滑动结束后在重新开启定时器让其自动轮播**

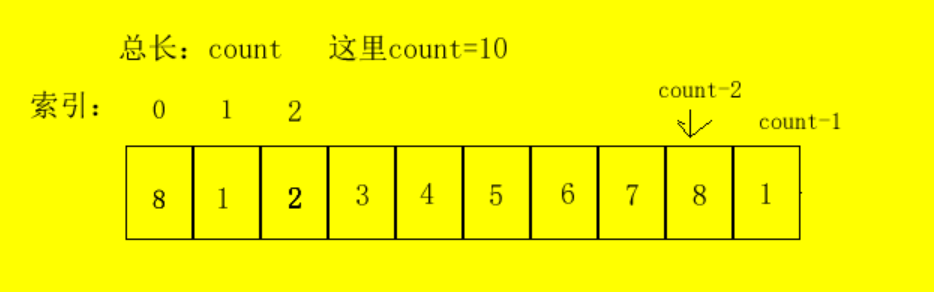
if(*Math*.abs(distanceX) > 100){  
 /\*判断滑动的方向\*/  
 if(distanceX > 0){//上一张  
 index--;  
 }  
 else{ //下一张  
 index++;  
 }  
 /\*翻页\*/  
 imgBox.style.transition="left 0.5s ease-in-out";  
 imgBox.style.left=-index\*bannerWidth+"px";  
}  
else if(*Math*.abs(distanceX) > 0){ //得保证用户确实进行过滑动操作  
 /\*回弹\*/  
 imgBox.style.transition="left 0.5s ease-in-out";  
 imgBox.style.left=-index\*bannerWidth+"px";  
}  
//重新开启定时器  
startTime();

## 5.4 20-移动web-轮播图-添加过渡结束的监听

1. **索引index和count关系**

总长为count由于索引index是从0开始计算，最后一张的索引就和count相差1

1. 最后一张index=count-1
2. Index=0和count-2也是同一张



1. **过渡结束事件的兼容性写法**

webkitTransitionEnd

imgBox.addEventListener("webkitTransitionEnd",function(){  
 /\*如果到了最后一张(count-1)，回到索引1\*/  
 /\*如果到了第一张(0)，回到索引count-2\*/  
 if(index==count-1){  
 index=1;  
 /\*清除过渡\*/  
 imgBox.style.transition="none";  
 /\*设置偏移\*/  
 imgBox.style.left=-index\*bannerWidth+"px";  
 }  
 else if(index==0){  
 index=count-2;  
 /\*清除过渡\*/  
 imgBox.style.transition="none";  
 /\*设置偏移\*/  
 imgBox.style.left=-index\*bannerWidth+"px";  
 }  
});