



小D课堂

愿景："让编程不在难学，让技术与生活更加有趣" 更多课程请访问xdclass.net

第一章 前期准备

第1节 课程简介及项目实战效果演示

第2节 分析页面结构以搭建页面架构



小D课堂

愿景："让编程不在难学，让技术与生活更加有趣" 更多课程请访问xdclass.net

第二章 搜索框及商品分类

第1节 初始化样式以及顶部栏布局

简介：讲解页面的先期初始化样式以及顶部栏的布局

初始化样式

```
* {  
  margin: 0;  
  padding: 0;  
}  
  
body,  
button,  
input {  
  font: 12px arial;  
}  
  
li {  
  list-style: none;  
}  
  
a {  
  color: #666;  
  text-decoration: none;  
}  
  
input,
```

```
button {
  border: none;
  outline: none;
}

body {
  background-color: #f8fafc;
}

.w {
  width: 1190px;
  margin: 0 auto;
}

.clearfix {
  *zoom: 1;
}

.clearfix::after {
  content: "";
  display: block;
  height: 0;
  clear: both;
}
```

第2节 详解搜索框下拉文字提示

简介：讲解搜索框的布局以及下拉文字提示的实现

- 监听按钮松开事件，显示下拉提示

```
searchInput.addEventListener("keyup", showKeyword, false);
```

- 监听失去焦点事件，隐藏下拉提示

```
searchInput.addEventListener("blur", hideKeyword, false);
```

- 监听获取焦点事件，显示下拉提示

```
searchInput.addEventListener("focus", showKeyword, false);
```

第3节 商品分类轮廓布局及原理分析

简介：讲解商品分类的布局以及实现原理

- 实现原理：
 - 利用绝对定位定位二级分类
 - 利用hover属性控制二级分类的显示和隐藏

第4节 详解二级分类实现(一)

第5节 详解二级分类实现(二)

简介：讲解二级分类的布局以及控制二级分类的显示隐藏

定位+hover属性

```
.nav-list {  
  position: relative;  
}  
.nav-list .item-menu {  
  display: none;  
  position: absolute;  
  left: 200px;  
  top: 0;  
  width: 900px;  
  min-height: 500px;  
  background-color: #fff;  
  box-shadow: 2px 2px 9px rgba(0, 0, 0, .2);  
  z-index: 999;  
}  
.list-item:hover .item-menu {  
  display: block;  
}
```

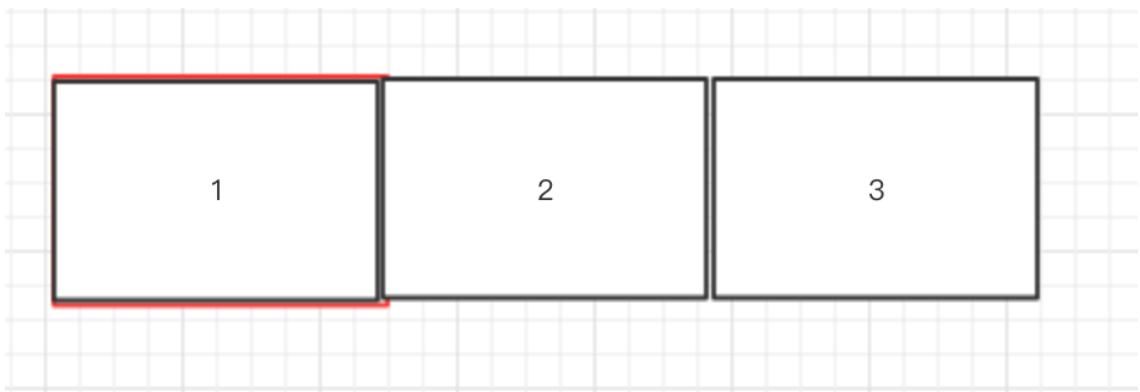


第三章 轮播图实现

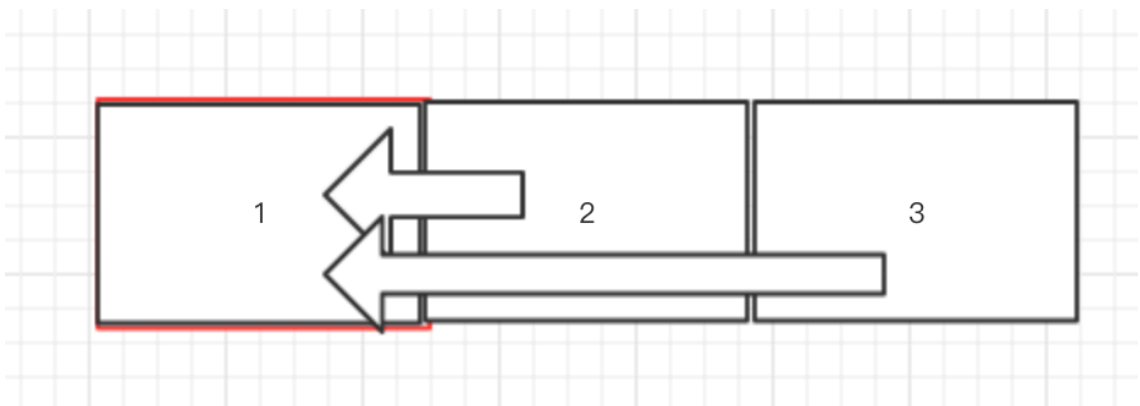
第1节 深度讲解轮播图布局及原理

简介：讲解轮播图的实现原理以及布局

- 轮播图实现原理
 - 将图片横向排列



- 将图片平移到容器里面



- 利用opacity属性控制图片的显示和隐藏

第2节 轮播图箭头按钮切换

简介：讲解如何使用按钮切换图片

- offsetWidth是元素的width+左右的padding+左右的border
- 动态添加opacity和translateX属性

```

for (var i = 0; i < swiperItem.length; i++) {
    if (index == i) {
        swiperItem[i].style.opacity = 1;
    } else {
        swiperItem[i].style.opacity = 0;
    }
    swiperItem[i].style.transform = "translateX(" + (-i *
swiperItem[0].offsetWidth) + "px)";
}

```

- 重置opacity为0，将当前索引的轮播图的opacity设置为1,达到更换图片的效果

```

function changeImg() {
    for (var j = 0; j < swiperItem.length; j++) {
        swiperItem[j].style.opacity = 0;
    }
    swiperItem[index].style.opacity = 1;
}

```

第3节 讲解轮播图循环播放和自动播放

简介：讲解轮播图的循环播放功能和自动播放功能

- 循环播放,索引超出最后一张时重新设置为0(第一张图片)，索引小于0时，就设置为最后一张的索引

```

if (index < 0) {
    index = swiperItem.length - 1;
} else if (index > swiperItem.length - 1) {
    index = 0;
}

```

- 自动播放，setinterval定时器

```
timer = setInterval(function(){
    index++;
    changeImg();
}, 3000);
```

第4节 详解鼠标经过停止播放和鼠标离开自动播放

简介：如何设置当鼠标经过时停止播放

- 元素监听鼠标经过mouseover事件

```
swiper.addEventListener("mouseover",function(){
    clearInterval(timer);
},false);
```

- 元素立案鼠标离开mouseout事件

```
swiper.addEventListener("mouseout",function(){
    autoChange();
},false)
```

第5节 轮播图底部圆点功能实现

简介：如何实现底部圆点以及点击切换图片功能

- 设置对应索引的圆点的激活状态

//设置圆点激活状态

```
function setIndicatorOn(){
    for(var i = 0;i < indicators.length;i++) {
        indicators[i].classList.remove("on");
    }
    indicators[index].classList.add("on");
}
```

- 给圆点添加点击事件切换图片

```
for(var k = 0;k < indicators.length;k++){
    indicators[k].onclick = function(){
        clearInterval(timer);
        var clickIndex = parseInt(this.getAttribute("data-index"));
        index = clickIndex;
        changeImg();
    }
}
```



小D课堂 愿景："让编程不在难学，让技术与生活更加有趣" 更多课程请访问xdclass.net

第四章 秒杀专区

第1节 讲解秒杀专区基本布局

简介：讲解秒杀专区的布局

- transform: scale缩放图片大小

```
.ms-item-lk:hover .ms-item-img {
    transform: scale(1.1);
    transition:transform .5s ease;
}
```

- 进度条实现方式

Html

```
<span class="progress"><span class="progress-bar"></span></span>
```

CSS

```
.ms-item-buy .progress {  
  display: inline-block;  
  width: 100px;  
  height: 10px;  
  background-color: #ddd;  
  border-radius: 5px;  
  margin: 5px 5px 0 0;  
}  
.ms-item-buy .progress .progress-bar {  
  display: block;  
  width: 30%;  
  height: 10px;  
  background-color: #e63e3e;  
  border-radius: 5px 0 0 5px;  
}
```

第2节 分析秒杀倒计时功能

简介：详细讲解秒杀专区倒计时功能的实现

- 运用定时器来实现倒计时功能

```
var ms_timer = setInterval(function () {  
  if (ms_time < 0) {  
    clearInterval(ms_timer);  
  } else {  
    calTime(ms_time);  
    ms_time--;  
  }  
}, 1000);
```

- 计算时分秒


```
function calTime(time) {
    var hour = Math.floor(time / 60 / 60); //小时
    var minutes = Math.floor(time / 60 % 60); //分钟
    var seconds = Math.floor(time % 60); //秒
    hour =formatTime(hour);
    minutes = formatTime(minutes);
    seconds = formatTime(seconds);
    document.getElementsByClassName("cd_hour")[0].innerHTML = hour;
    document.getElementsByClassName("cd_minute")[0].innerHTML = minutes;
    document.getElementsByClassName("cd_second")[0].innerHTML = seconds;
}
```

- 格式化时间

```
function formatTime(t) {
    if (t < 10) {
        t = "0" + t;
    }
    return t;
}
```

第3节 热门专区模块实现

简介：详细讲解热门专区模块布局

- 添加hover属性和box-shadow

```
.saleList .bs-item:hover {
    box-shadow: 0 0 20px rgba(0, 0, 0, .2);
}
```

第4节 推荐喜欢列表布局

简介：讲解推荐喜欢列表的核心布局知识点

第5节 网页必备底部版权信息

简介：讲解网页底部版权信息的布局



小D课堂 愿景："让编程不在难学，让技术与生活更加有趣" 更多课程请访问xdclass.net

第五章 模拟接口请求及悬浮菜单

第1节 讲解本地json数据编写及接口封装

简介：详细讲解如何使用本地json数据模拟接口请求以及接口封装

- readyState五种状态
 - 0, 请求未初始化, 还没创建xmlhttprequest对象
 - 1, 建立服务器的连接, 调用open
 - 2, 发送请求, send()
 - 3, 请求正在处理
 - 4, 请求已完成且相应结束
- status三个状态
 - 200 "OK" 请求成功
 - 403 "Forbidden" 服务器禁止请求
 - 404 "not found" 找不到这个请求

第2节 详解搜索模块模拟接口对接及防抖函数

简介：讲解搜索模块下拉提示模拟接口请求和防抖函数的实现

- 防抖函数

```
//防抖函数
function debounce(fn,delay){
    var handle;
    return function(){
        clearTimeout(handle);
        handle = setTimeout(function(){
            fn();
        },delay)
    }
}
```

第3节 深度讲解轮播图接口请求及功能完善

简介：详细讲解轮播图接口对接

- 获取轮播图列表，动态插入轮播图容器

```
//获取轮播图列表
function getBannerList(){
    ajax("get","swiper.json",function(res){
        console.log(res);
        if(res.code == 0){
            var data = res.data;
            //轮播列表循环插入容器
            var swiper = document.getElementById("swiper");
            var str = "";
            for(var i = 0;i< data.length;i++){
                str += '<li class="swiper-item"><a href="#"></a></li>';
            }
            swiper.innerHTML = str;
            //圆点列表循环插入容器
```

```

var indicators = document.getElementById("indicators");
var str2 = "";
for(var i = 0;i< data.length;i++){
    if(i == 0){
        str2 += '<div class="on indicator" data-index="'+i+'"></div>'
    }else{
        str2 += '<div class="indicator" data-index="'+i+'"></div>';
    }
}
indicators.innerHTML = str2;
bannerOption();
}
},true)
}

```

第4节 秒杀专区及热门专区接口对接

简介：详细讲解秒杀专区和热门专区数据格式编写以及接口对接

- 秒杀专区数据对接

```

//获取秒杀专区数据
function msApi() {
    ajax("get", "miaosha.json", function (res) {
        // console.log(res);
        if (res.code == 0) {
            var cd_time = res.data.ms_time;
            var good_list = res.data.goods_list;
            countdown(cd_time);
            var miaoshaList = document.getElementById("miaoshaList");
            var str = "";
            for (var i = 0; i < good_list.length; i++) {
                str += '<div class="ms-item">' +
                    '<a href="#" class="ms-item-lk">' +
                    '' +

```

```

        '<p class="ms-item-name">' + good_list[i].name + '</p>' +
        '<div class="ms-item-buy">' +
        '<span class="progress"><span class="progress-bar"
style="width:' + good_list[i].progress + '%"></span></span>' +
        '<span class="buy-num">已抢' + good_list[i].progress + '%
</span>' +
        '</div>' +
        '<div class="ms-item-price clearfix">' +
        '<span class="cm-price new-price">¥' + good_list[i].new_price
+ '</span>' +
        '<span class="cm-price origin-price">¥' +
good_list[i].old_price + '</span>' +
        '</div>' +
        '</a>' +
        '</div>';
    }
    miaoshaList.innerHTML = str;
}
}, true);
}

```

- 热门专区数据对接

//获取热卖单品列表

```

function hotSaleApi() {
    ajax("get", "hotsale.json", function (res) {
        console.log(res);
        if (res.code == 0) {
            var list = res.data;
            var hotSaleList = document.getElementById("hotSaleList");
            var str = "";
            for (var i = 0; i < list.length; i++) {
                str += '<li class="bs-item item">' +
                    '<a href="">' +
                    '' +
                    '<p class="title">'+list[i].name+'</p>' +
                    '<div class="line-2 clearfix">' +
                    '<span class="comment">评论<em>'+list[i].commentNum+'</em></span>'
+
                    '<span class="collect">收藏<em>'+list[i].collectNum+'</em></span>'
+
                    '</div>' +
                    '<div class="line-3">' +
                    '<span class="strong">'+list[i].new_price+'</span>' +
                    '<span class="price-through">¥'+list[i].old_price+'</span>' +
                    '<span class="sell">月销'+list[i].saleNum+'笔</span>' +
                    '</div>' +

```

```

        '</a>' +
        '</li>';
    }
    hotSaleList.innerHTML = str;
}
}, true)
}

```

第5节 如何实现推荐喜欢列表加载更多功能

简介：详细讲解推荐喜欢列表数据对接以及加载更多功能实现

- 加载更多功能

```

document.getElementById("loadMore").onclick = function () {
    var liNode = document.createElement("li");
    liNode.setAttribute("class", "like-item item");
    var liContent = `<a href="">
        
    </a>
    <p class="title">分布式无线路由器</p>
    <div class="line-3">
        <span class="strong">1100</span>
        <span class="sell">月销3000笔</span>
    </div>
    <a href="" class="item-more">发现更多相似的宝贝</a>`;
    liNode.innerHTML = liContent;
    document.getElementById("gl").appendChild(liNode);
}

```

- es6语法 `` 字符串拼接, \${params} 接入参数

```

`<li class="like-item item">
    <a href="">
        
    </a>
    <p class="title">${list[i].name}</p>
    <div class="line-3">
        <span class="strong">${list[i].new_price}</span>
        <span class="sell">月销${list[i].saleNum}笔</span>
    </div>
    <a href="" class="item-more">发现更多相似的宝贝</a>
</li>`

```

第6节 提高用户体验之悬浮菜单实现

简介：详细讲解悬浮功能实现和滚动监听菜单

- 滚动监听菜单

```

window.onscroll = function(){
    scrollShowBtn();
    var winPos = document.documentElement.scrollTop ||
document.body.scrollTop; //获取滚动的距离
    var hotSale = document.getElementById("hotsale"); //获取热卖专区元素
    var hotHeight = hotSale.offsetTop + hotSale.offsetHeight; //猜你喜欢之前的总高度

    if(winPos < hotSale.offsetTop){
        addOn(0);
    }else if(winPos >= hotSale.offsetTop && winPos < hotHeight){
        addOn(1)
    }else{
        addOn(2);
    }
}

//添加菜单激活状态

```

```
function addOn(index){
    var tool = document.getElementsByClassName("tool")[0];
    var toolItem = tool.getElementsByTagName("a");
    for(var i =0;i < toolItem.length;i++){
        toolItem[i].classList.remove("on");
    }
    toolItem[index].classList.add("on");
}
```

- 返回顶部从快到慢的效果

```
//返回顶部
function goTop(){
    var topTimer = setInterval(function(){
        var scrollTop = document.documentElement.scrollTop ||
document.body.scrollTop;
        var iSpeed = Math.floor(-scrollTop/8);
        if(scrollTop == 0){
            clearInterval(topTimer);
        }
        document.documentElement.scrollTop = document.body.scrollTop =
scrollTop + iSpeed;
    },30)
}
```

第7节 界面优化及阿里矢量图标引入

简介：讲解阿里矢量图标引入

- 阿里矢量图标引入
 - 引入iconcss文件

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

- 引用对应图片class类

```
<span class="iconfont icon-leijiayongdianqix"></span>
```




第六章 课程总结及前端学习路线

第1节 课程总结及常见问题解决方法

- 怎么让不定宽高的div，垂直水平居中？
 - flex布局

```
给父元素设置
display: flex;
justify-content: center;
align-items: center;
```

- transform属性

```
利用绝对定位以及translate属性
{
  position: absolute;
  left: 50%;
  top: 50%;
  transform: translate(-50%, -50%)
}
```

- 表格属性table-cell

```
父元素设置
{
  display: table-cell;
  text-align: center;
  vertical-align: middle;
}

子元素设置display: inline-block
```

- CSS隐藏元素的几种方式及区别
 - display: none, 元素为彻底消失，原有空间被其他元素占有

- Visibility:hidden与display:none的区别：原有空间还是会存在页面，无法触发其点击事件

适用于元素隐藏后不希望页面布局会发生变化的场景

- Opacity:0 透明设置为零让元素看不见，原有空间也是保留下来的，但是它是可以触发其点击事件
- CSS控制文字，超出部分显示省略号
 - 单行文本的溢出显示省略号

```
text-overflow:ellipsis;
overflow:hidden;
white-space:nowrap;
```

- 多行文本溢出显示省略号

```
display: -webkit-box;
-webkit-box-orient: vertical;
-webkit-line-clamp:2;
overflow: hidden;
text-overflow: ellipsis;
```

- CSS外边距合并问题(外边距塌陷)
 - 子元素与父元素的外边距问题

父元素上设置

- 1、overflow: hidden
- 2、border:1px solid transparent;
- 3、padding:1px;
- 4、display:inline-block

- 兄弟元素外边距合并问题

- 1、float: left
- 2、display: inline-block

第2节 通往高级前端工程师之完整学习路线

- 第一阶段：掌握基础

html+css+html5+css3+javascript+jquery

- 第二阶段：进阶

vue.js、react.js、angular.js、bootstrap、css预编译器(sass、less等)、(webpack/gulp)、es6以上版本、

uniapp、微信小程序、项目管理git、svn

- 第三阶段：高级

nodejs (express/koa) 教程、mysql教程、linux服务器

