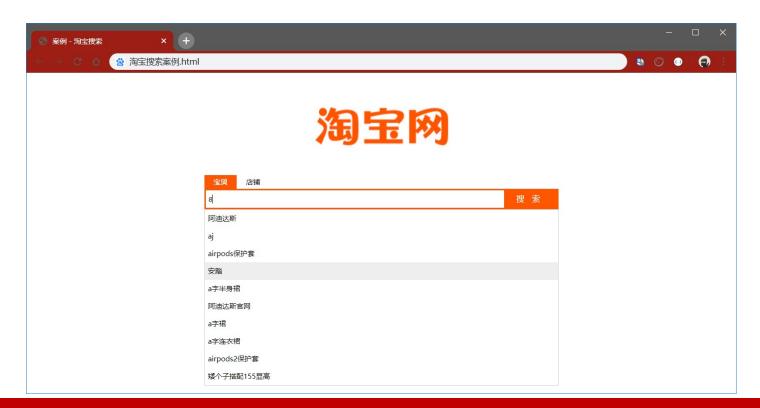




◆ 案例-淘宝搜索



3.1 要实现的UI效果





3.2 获取用户输入的搜索关键词

为了获取到用户每次按下键盘输入的内容,需要监听输入框的 keyup 事件,示例代码如下:

```
// 监听文本框的 keyup 事件
$('#ipt').on('keyup', function() {
 // 获取用户输入的内容
 var keywords = $(this).val().trim()
 // 判断用户输入的内容是否为空
 if (keywords.length <= 0) {</pre>
   return
 // TODO: 获取搜索建议列表
})
```



3.3 封装getSuggestList函数

将获取搜索建议列表的代码, 封装到 getSuggestList 函数中, 示例代码如下:

```
function getSuggestList(kw) {
 $.ajax({
    // 指定请求的 URL 地址, 其中, q 是用户输入的关键字
    url: 'https://suggest.taobao.com/sug?q=' + kw,
    // 指定要发起的是 JSONP 请求
    dataType: 'jsonp',
    // 成功的回调函数
    success: function(res) { console.log(res) }
 })
```



3.4 渲染建议列表的UI结构

1. 定义搜索建议列表



3.4 渲染建议列表的UI结构

2. 定义模板结构

```
<!-- 模板结构 -->
<script type="text/html" id="tpl-suggestList">
{{each result}}

<div class="suggest-item">{{$value[0]}}</div>
{{/each}}
</script>
```



3.4 渲染建议列表的UI结构

3. 定义渲染模板结构的函数

```
// 渲染建议列表
function renderSuggestList(res) {
  // 如果没有需要渲染的数据, 则直接 return
  if (res.result.length <= 0) {</pre>
     return $('#suggest-list').empty().hide()
   // 渲染模板结构
   var htmlStr = template('tpl-suggestList', res)
  $('#suggest-list').html(htmlStr).show()
```



3.4 渲染建议列表的UI结构

4. 搜索关键词为空时隐藏搜索建议列表

```
$('#ipt').on('keyup', function() {
  // 获取用户输入的内容
  var keywords = $(this).val().trim()
  // 判断用户输入的内容是否为空
  if (keywords.length <= 0) {</pre>
     // 如果关键词为空,则清空后隐藏搜索建议列表
     return $('#suggest-list').empty().hide()
  getSuggestList(keywords)
})
```





3.5 输入框的防抖

3. 实现输入框的防抖

```
var timer = null
                                 // 1. 防抖动的 timer
function debounceSearch(keywords) { // 2. 定义防抖的函数
  timer = setTimeout(function() {
  // 发起 JSONP 请求
  getSuggestList(keywords)
  }, 500)
$('#ipt').on('keyup', function() { // 3. 在触发 keyup 事件时, 立即清空 timer
  clearTimeout(timer)
  // ...省略其他代码
  debounceSearch (keywords)
})
```



3.6 缓存搜索的建议列表

1. 定义全局缓存对象

```
// 缓存对象
var cacheObj = {}
```



3.6 缓存搜索的建议列表

2. 将搜索结果保存到缓存对象中



3.6 缓存搜索的建议列表

3. 优先从缓存中获取搜索建议

```
// 监听文本框的 keyup 事件
$('#ipt').on('keyup', function() {
  // ...省略其他代码
  // 优先从缓存中获取搜索建议
  if (cacheObj[keywords]) {
     return renderSuggestList(cacheObj[keywords])
  // 获取搜索建议列表
  debounceSearch (keywords)
```