

回退方法：

<

家政头条

...

@click="goHistory":

```
goHistory: function () {  
  this.$router.go(-1);  
},
```

功能：回退一步，返回上一步操作

优化：（分享链接后直达文章详情，没有window.history对象，判断后决定跳转首页还是回退一步）

```
goHistory: function () {  
  if(window.history.length <= 1){  
    this.$router.push({path: "/main"});  
  }else{  
    this.$router.go(-1);  
  }  
},
```

官方文档：

`router.go(n)`

这个方法的参数是一个整数，意思是在 history 记录中向前或者后退多少步，类似 `window.history.go(n)`。

例子

```
// 在浏览器记录中前进一步，等同于 history.forward()  
router.go(1)  
  
// 后退一步记录，等同于 history.back()  
router.go(-1)  
  
// 前进 3 步记录  
router.go(3)  
  
// 如果 history 记录不够用，那就默默地失败呗  
router.go(-100)  
router.go(100)
```

JS

路由跳转：

①：声明式 `<router-link :to="...">`

`<router-link :to="{path: '/articleDetail', query: {articleId: 'WZ18012600000217'}}">` 点击跳转 `</router-link>`

②：编程式 `router.push(...)`

`this.$router.push({path: "/articleDetail", query: {articleId: articleId}});`

路由传参：

①：由主页文章列表向文章详情跳转，带参数 (articleId)

- 定义方法： `@click="articleDetail(item.id)"`
- 方法中跳转路由：

```
articleDetail: function (articleId) {  
    this.$router.push({path: "/articleDetail", query: {articleId: articleId}});  
},
```

- 目标页面得到参数：

```
let articleId = this.$route.query.articleId;
```

②：（之前用过，不太好用，所以放弃）

- 命名路由： `name: '/articleDetail/:articleId', params: { articleId: 123 }, component: ArticleDetail`
- 跳转路由： `router.push({ name: '/articleDetail', params: { articleId: 123 } })`
- 目标页面得到参数：`props: ['articleId'],`

注意：如果提供了 path, params 会被忽略。

```
const userId = 123
```

```
router.push({ name: 'user', params: { userId } }) // -> /user/123
```

```
router.push({ path: `/user/${userId}` }) // -> /user/123
```

// 这里的 params 不生效

```
router.push({ path: '/user', params: { userId } }) // -> /user
```

图片绑定：

`<img`

```
v-bind:src="(userInfo.photoUrl? userInfo.photoUrl :  
'//img.1000.com/ehome/dev/P00000017/1514424805951.jpg')+ '?x-oss-  
process=image/resize,m_lfit,h_35,w_35' "  
alt=" " />
```

v-html的用处:

媒体项目中，文章详情（正文内容）是由百度编辑器自动生成，传给数据库一串字符串，如：

```
contentUrl:"<p>编辑器测试正文</p><h1>编辑器测试正文</h1><h2>编辑器测试正文</h2><h3>编辑器测试正文</h3><p>本地上传<br/></p><p></p>  
<p>网络图片</p><p></p>"
```

如果以普通标签展示如上，加上**v-html="contentUrl"**就会正常展示。

监听： (watch)

watch

- 类型： `{ [key: string]: string | Function | Object | Array }`
- 详细：

一个对象，键是需要观察的表达式，值是对应回调函数。值也可以是方法名，或者包含选项的对象。Vue 实例将会在实例化时调用 `$watch()`，遍历 watch 对象的每一个属性。

在文章详情页中，相关推荐的文章列表需要跳转文章详情页，就可以用watch监听新老数据实现。

文章正文中图片加样式问题:

//解决图片加样式问题

```
let content = that.articleDetail.contentUrl;
if(content.indexOf("<img")>=0) {
    let _content = content;
    let reg = /<img.*?(?:>|\>)/g;
    let contentList = content.split(reg);
    for (let item of contentList) {
        _content = _content.replace(item, "");
    }
    _content = _content.match(reg);
    for(let item of _content)
    {
        if(item.indexOf("style=")>=0) {
            let newImg = item.replace(/;"\>/g, ' ;max-width:100%"/>');
            content = content.replace(item, newImg);
        } else {
            let newImg = item.replace(/\>/g, ' style="max-width:100%"/>');
            content = content.replace(item, newImg);
        }
    }
}
that.content = content;
```

绑定样式：

问题情境：要求处于当前栏目时，栏目显示样式。

微信端测试标题1

热 牛人牛评 2018-01-26



dddddddddddddddddddddd
d

热 牛人牛评 2017-12-28



呃呃呃呃呃呃呃呃呃呃呃呃呃呃呃
呃呃呃呃

热 牛人牛评 2017-12-27



家政头条标题 家政头条标题 家
政头条标题 家政头条标题 家政

热 牛人牛评 2017-12-27



```
<a @click="queryArticleByColumn(item.value)" v-bind:class="[item.value ==  
columnId ? 'active' : '']">
```

思路：在click事件触发时，将列表的columnId 赋值给本页的columnId，展示时进行v-bind:class= “ ” 的判断是否展示class。

可以参照官方文档Class 与 Style 绑定一章。

<https://cn.vuejs.org/v2/guide/class-and-style.html>

#对象语法：

```
v-bind:class="{active:columnId==item.value}"
```

#数组语法：

```
v-bind:class="[item.value == columnId ? 'active' : ''] "
```