

# 项目笔记

## 视口单位

在PC端，视口指的是在PC端，指的是浏览器的可视区域；

而在移动端，它涉及3个视口：Layout Viewport（布局视口），Visual Viewport（视觉视口），Ideal Viewport（理想视口）。

- 1. vw：1vw等于视口宽度的1%。
- 2. vh：1vh等于视口高度的1%。
- 3. vmin：选取vw和vh中最小的那个。
- 4. vmax：选取vw和vh中最大的那个。

Img用100%填充浏览器放大缩小会变形，如果要随浏览器的变化而变化，就要用浏览器宽度减去左右两边其他元素的宽度。

```
width: calc(100vh-450px);
height: calc(100vh - 60px);
```

## js事件

### 鼠标事件

事件	触发
onclick	点击时
ondblclick	双击时
onmousedown	按下左键释放之前
onmouseup	在页面元素上移动鼠标
onmouseout	指针移出元素范围
onmouseover	第一次移入指定元素范围
onmouseup	释放按下的左键时发生
ondrag	元素被拖动时
ondragstart	在拖动操作开端运行
ondragend	拖动操作末端运行
ondragenter	当元素元素已被拖动到有效拖放区域时
ondragleave	当元素离开有效拖放目标时运行
ondragover	当元素在有效拖放目标上正在被拖动时

## 键盘事件

onkeypress	用户按下一键未释放时
onkeydown	单击键盘时
onkeyup	键盘按下再放开时发生

## 表单事件

事件	触发
onsubmit	用户提交表单时发生
onchange	在域的内容改变时发生
onreset	表单数据被重置时发生
onselect	用户从文本框中选择文本时发生
onfocus	获得焦点时
onblur	失去焦点时触发

## 读取事件

事件	触发
onabort	用户停止浏览器继续下载图像时触发
onerror	当浏览器载入网页或图像发生错误时
onload	通常用于 body元素，在页面完全载入后触发(包括图片、css文件等等。)
onresize	窗口或框架被调整大小时发生。
onunload	用户离开页面时发生

## onload与onpageshow事件区别

onpageshow 事件类似于 onload 事件，onload 事件在页面第一次加载时触发， onpageshow 事件在每次加载页面时触发，即 onload 事件 在页面 从浏览器缓存中读取时不触发。

## 其它事件

事件	触发
onscroll	在元素滚动条在滚动时触发

## addEventListener()

addEventListener(event, function, useCapture)

1. 参数event必填，表示监听的事件，例如 click, touchstart 等，不加前缀on的事件
2. 参数 function必填，表示事件触发后调用的函数，可以是外部定义函数，也可以是匿名函数。不带参数
3. 参数 useCapture选填，填true或者false，用于描述事件是冒泡还是捕获触发，true表示捕获，默认false表示冒泡。

这里function参数传的是匿名函数,没办法用removeEventListener解除监听

```
window.addEventListener('resize', function() {  
  console.log('resize')  
}, false)
```

这里function参数传的不是匿名函数,可以用removeEventListener解除监听

```
function resizeFn(e) {  
  console.log('resize')  
  console.log(e)  
}  
window.addEventListener('resize', resizeFn, false)  
// 解除上面绑定的事件监听  
window.removeEventListener('resize', resizeFn, false)
```

## git拉取代码

### 下载代码

1. 创建文件夹在该文件夹下右键git base
2. git init 使用当前目录作数据仓库
3. 创建并进入分支 git checkout -b feat-yaotingting
4. 打开git仓库，复制链接。
5. 下载代码git clone + url
6. 安装依赖npm i
7. git pull 下载远程代码并合并
8. git add . 添加文件到暂存区
9. git commit -m " " 将暂存区的内容添加到仓库中，记得选择feat,查看代码规范
10. git push 上传到自己分支
11. git status  
看看上传的情况，nothing表示都上传了
12. git checkout feat-0625 进入主分支
13. git merge feat-yaotingting 合并分支命令:
14. git push origin feat-0625 上传远程代码并合并
15. git checkout feat-yaotingting 继续写代码

## mount无法读取data中的数据

原因就是：定时器的this是指向 window的。解决方法：

1. 用箭头函数：箭头函数中的this指向是固定不变（定义函数时的指向），在vue中指向vue;
2. 使用 var that = this ,就可以正常调用了。

## 人工标注项目添加页面

---

1. TaskManage下创建vue文件名同文件名相同。注意for循环加key，避免图片找不到地址还有引入不存在的文件。
2. 在router.js中找到正确路径下添加路由
3. 在搜索框搜查看找到,taskList.vue添加按钮。名字和下面的method对应

```
<el-button  
  type="text"  
  size="small"  
  @click="goToMarking(scope.row.id)"  
>查看  
</el-button>
```

在method中添加，注意路径：

```
goToMarking(id) {  
  this.$router.push({  
    path: 'marking',  
    query: { id: id },  
  })  
},
```

## 明日计划

---

1. 把剩余页面搭建完
2. 学习项目代码规范