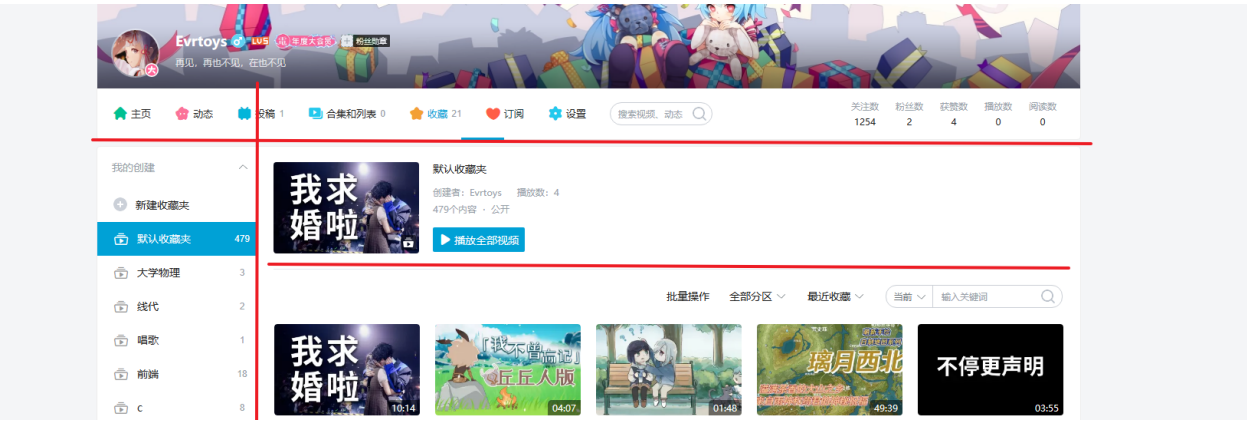
个人主页

布局

> ⚠️ 在flex布局下，如果右侧的内容的宽度大于盒子宽度可能导致flex错乱，这时给右侧盒子添加`width:0px;`属性，让这个盒子的width属性由flex决定




```
<div class="collections w">
    <div class="head">我的收藏夹<i>数量</i></div>
    <div class="collectionList">
        <div class="collection">
            <div class="video">
                <video src=""></video>
            </div>
            <div class="description"></div>
        </div>
    </div>
</div>
```

收藏模块

实现收藏动作

> 在登录的状态下收藏才可用，收藏的视频要知道视频的VideoObj

- 收藏要区分：视频，用户，不同的用户的收藏是不一样的,用localStorage实现需要储存在以用户名命名的VieoObj的数组中

- 收藏之后，这个视频对于这个人就是已经被收藏的状态，所以下次打开也是收藏的，所以，加载视频的时候要判断收藏的状态，如果当前视频在这个数组里面那么就是已经收藏的状态`JSON.parse(localStorage.getItem(`${localStorage.getItem('loginUser')}+Info`))`

  页面视频的结构

```
<div class="collection">
     <div class="video">
         <video src=""></video>
      </div>
     <div class="description">avata</div>
</div>
```

展示收藏视频

> 注意保留排序的接口，就是说渲染函数拿到的是一个数组，排序是用数组的顺序来控制

排序

拖动排序实现

> drag事件

```
let dragList = document.querySelector('.content .edit ul')//还有一个所以要再来一次
let draggedElement;
let draggedOrder;
let animation = false;//标记做动画的过程，在这个过程中不能再次触发动画了
dragList.addEventListener('dragstart', (e) => {
    draggedElement = e.target
    draggedOrder = Array.from(draggedElement.parentNode.children).indexOf(draggedElement)
})

dragList.addEventListener('dragenter', (e) => {//只有进入的时候会执行
    e.preventDefault()
    let order = Array.from(e.target.parentNode.children).indexOf(e.target)
    //判断先后，执行动画，调换位置：
    if (e.target !== draggedElement && !animation) {
        if ((order > draggedOrder)) {
            animation = true
            draggedElement.classList.add("dragSortingDown")
            e.target.classList.add("dragSortingUp")
            e.target.addEventListener("animationend", () => {
                dragList.insertBefore(e.target, draggedElement)
                e.target.classList.remove("dragSortingUp")
                draggedElement.classList.remove("dragSortingDown")
                animation = false
                //更新拖动元素的下标
                draggedOrder = Array.from(e.target.parentNode.children).indexOf(draggedElement)

                return;
            })
        }
        else if ((order < draggedOrder)) {
            animation = true
            e.target.classList.add("dragSortingDown")
            draggedElement.classList.add("dragSortingUp")
            e.target.addEventListener("animationend", () => {
                dragList.insertBefore(draggedElement, e.target)
                e.target.classList.remove("dragSortingDown")
                draggedElement.classList.remove("dragSortingUp")
                animation = false
                //更新拖动元素的下标
                draggedOrder = Array.from(e.target.parentNode.children).indexOf(draggedElement)

                return;
            })
        }
    }
})

dragList.addEventListener('dragover', (e) => {
```

```
    e.preventDefault()
})
```

搜索

通过正则实现模糊搜索

js中正则表达式的使用

> 💡 搜索之后可以高亮关键词

更换头像

> 在个人主页上面的图片的头像处点击头像触发更换头像事件，post到服务器即可

上传视频

> 在没有服务端的情况下播放视频，方法一，input.value获取视频路径，直接通过读取本地视频来播放，方法二，创建临时url

> ⚠️ 用input.value拿到的不是真实的路径，而是一个fakepath

- 使用URL.createObjectURL()方法，可以获取一个表示视频内容的临时URL，这个URL可以用于在网页中显示视频，但是需要在使用后调用URL.revokeObjectURL()方法来释放资源**23**。例如：

```
<input type="file" id="file" onchange="showVideo(this.files[0])">
<video id="video"></video>
<script>
  function showVideo(file) {
    var url = URL.createObjectURL(file);
    var video = document.getElementById("video");
    video.src = url;
    video.onload = function() {
      URL.revokeObjectURL(url);
    };
  }
</script>
```

其中file = input.files[0]

渲染

通过封装这个video的信息，让他的对象和我们从服务器拿出来的一样，然后就可以直接用以前的方法来渲染

> ⚠️ **BUG：当没有收藏的时候**，要注意加载的时候跳出：
> `if (JSON.parse(localStorage.getItem(`${localStorage.getItem('loginUser')}Info`)) !== null)`
> `if (JSON.parse(localStorage.getItem(`${localStorage.getItem('loginUser')}Info`))[0] !== undefined)`

```
// 加载默认收藏夹
if (JSON.parse(localStorage.getItem(`${localStorage.getItem('loginUser')}Info`))[0] !== undefined) {
    document.querySelector('.myClt .collections .default .box').querySelector('.video video').setAttribute('src', `${JSON.parse(localStorag
}
```