

深圳大学实验报告

课程名称: web 开发与人机交互导论

实验项目名称: 实验 3 超链接与多媒体文件应用

学院: 计算机与软件学院

专业: 计算机科学与技术

指导教师: 李俊杰

报告人: 曹秦鲁 学号: 2023150203 班级: 国际班

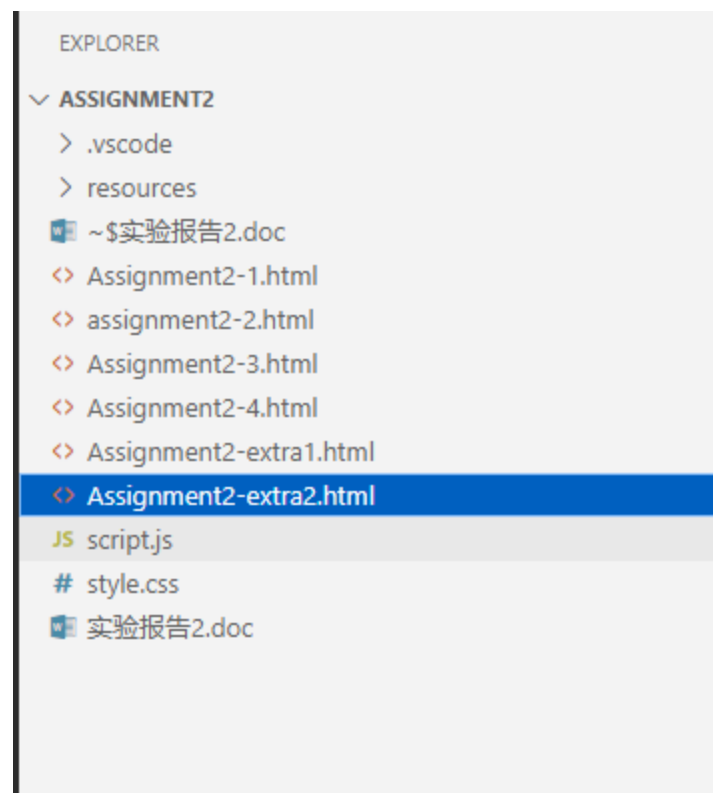
实验目的与要求：

通过本次作业，学生可以学习和掌握以下技能：

- **HTML 结构**：HTML 的基本结构和元素。
- **CSS 样式**：使用 CSS 定义全局样式、标题样式、页脚样式、容器样式和链接样式。
- **Flexbox 布局**：掌握 Flexbox 布局的基本用法，能够实现子元素的水平居中、换行和间距设置。

实验过程及内容：

所有项目：



项目 1 设计简易灯箱画廊

这个项目是一个简易的灯箱画廊网页，用户可以点击缩略图来查看大图。项目由三个主要文件组成：HTML 文件、CSS 文件和 JavaScript 文件。

代码截图

```
<> assignment2-1.html ● # style.css ● Settings JS script.js ●
<> assignment2-1.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>简易灯箱画廊</title>
7      <link rel="stylesheet" href="style.css">
8  </head>
9  <body>
10     <h1>简易灯箱画廊</h1>
11     <div class="gallery">
12         <a href="resources/image31.jpg" class="thumbnail">
13             
14         </a>
15         <a href="resources/image32.jpg" class="thumbnail">
16             
17         </a>
18         <a href="resources/image33.jpg" class="thumbnail">
19             
20         </a>
21         <!-- 添加更多缩略图 -->
22     </div>
23     <div class="lightbox" id="lightbox">
24         <img id="lightbox-img" src="" alt="Large Image">
25     </div>
26     <script src="script.js"></script>
27 </body>
28 </html>
```

代码解释：

将 css 单独建一个文件，使得项目具有良好的布局。

因为要点击图片后再显示灯箱，所以默认 display: none

z-index: 1000 赋予灯箱较高的层级，确保其显示在其他元素之上，不被遮挡

```
#lightbox {
    display: none; /* 初始状态下隐藏灯箱 */
    justify-content: center; /* 水平居中对齐 */
    align-items: center; /* 垂直居中对齐 */
    position: fixed; /* 固定定位 */
    bottom: 0; /* 灯箱定位在页面的下半部分 */
    left: 0;
    width: 100%; /* 灯箱宽度为 100% */
    height: 50%; /* 灯箱高度为 50% */
    background-color: rgba(0, 0, 0, 0.8); /* 半透明黑色背景 */
    z-index: 1000; /* 设置较高的层级 */
    overflow: hidden; /* 隐藏溢出内容 */
}
```

```
.gallery {
  display: flex;
  flex-wrap: wrap;
  gap: 10px;
  justify-content: center; /* 使画廊居中 */
}
```

通过 flex 布局使得画廊图片在不同的比例下都可以灵活展示，也就不会因为边框的移动遮挡住图片。

通过 javascript 脚本实现缩略图到大图的转变：

对每个缩略图元素添加一个点击事件监听器，等待用户的点击操作。

```
thumbnails.forEach(thumbnail => {
  thumbnail.addEventListener('click', function(event) {
    // ...
  });
});
```

```
const largeImageSrc = this.href;
```

使用 this.href 获取被点击的缩略图链接所指向的大图地址

CSS 文件：

```
1. /* style.css */
2. body {
3.   font-family: Arial, sans-serif;
4.   text-align: center; /* 使标题居中 */
5. }
6.
7. h1 {
8.   margin-top: 20px;
9.   font-size: 2em;
10.  color: #333;
11. }
12.
13. .gallery {
14.   display: flex;
15.   flex-wrap: wrap;
16.   gap: 10px;
17.   justify-content: center; /* 使画廊居中 */
```

```
18. }
19.
20. .thumbnail {
21.     position: relative;
22.     display: inline-block;
23. }
24.
25. .thumbnail img {
26.     width: 100px;
27.     height: 100px;
28.     object-fit: cover;
29.     cursor: pointer;
30. }
31.
32. .caption {
33.     position: absolute;
34.     bottom: 0;
35.     left: 0;
36.     width: 100%;
37.     background-color: rgba(0, 0, 0, 0.5);
38.     color: white;
39.     text-align: center;
40.     padding: 5px 0;
41.     font-size: 12px;
42. }
43.
44. .lightbox {
45.     position: fixed;
46.     bottom: 0;
47.     left: 0;
48.     width: 100%;
49.     background-color: rgba(0, 0, 0, 0.8);
50.     text-align: center;
51.     padding: 10px;
52.     display: none;
53. }
54.
55. .lightbox img {
56.     max-width: 100%;
57.     height: auto;
58. }
```

java script 代码:

```

1. // script.js
2. document.addEventListener('DOMContentLoaded', function() {
3.     const thumbnails = document.querySelectorAll('.thumbnail');
4.     const lightbox = document.getElementById('lightbox');
5.     const lightboxImg = document.getElementById('lightbox-img');
6.
7.     thumbnails.forEach(thumbnail => {
8.         thumbnail.addEventListener('click', function(event) {
9.             event.preventDefault();
10.            const largeImageSrc = this.href;
11.            lightboxImg.src = largeImageSrc;
12.            lightbox.style.display = 'flex'; // 使用 flex 布局显示
13.        });
14.    });
15.
16.    lightbox.addEventListener('click', function() {
17.        lightbox.style.display = 'none';
18.    });
19. });

```

结果展示：



项目 2 设计支持音频、视频播放的网页

代码截图

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>音频和视频播放示例</title>
7      <link rel="stylesheet" href="style.css">
8  </head>
9  <body>
10     <h1>明月几时有</h1>
11
12     <h2>歌词</h2>
13     <div class="lyrics">
14         <p>明月几时有？把酒问青天。</p>
15         <p>不知天上宫阙，今夕是何年。</p>
16         <p>我欲乘风归去，又恐琼楼玉宇，高处不胜寒，起舞弄清影，何似在人间。</p>
17         <p>转朱阁，抵绮户，照无眠。</p>
18         <p>不应有恨，何事偏向别时圆。</p>
19         <p>人有悲欢离合，月有阴晴圆缺，此事古难全。</p>
20         <p>但愿人长久，千里共婵娟。</p>
21     </div>
22
23     <h2>音频播放</h2>
24     <audio controls>
25         <source src="resources/embed/蔡琴明月几时有.mp3" type="audio/mpeg">
26         您的浏览器不支持音频元素。
27     </audio>
28
29     <h2>视频播放</h2>
30     <video controls width="600">
31         <source src="resources/puppy.mp4" type="video/mp4">
32         您的浏览器不支持视频元素。
33     </video>
34
35     <script src="script.js"></script>
36 </body>
37 </html>
```

代码解释：

Lyrics css 代码：

使得歌词文本具有美感：

通过背景颜色和阴影使得歌词框突出，与其他背景分别出来

```

1.  /* 歌词样式 */
2.  .lyrics {
3.      font-family: 'KaiTi', 'STKaiti', serif; /* 设置字体为楷体 */
4.      font-size: 1.2em; /* 设置字体大小 */
5.      line-height: 1.8; /* 设置行高 */
6.      color: #555; /* 设置文本颜色 */
7.      margin: 20px auto; /* 设置上下外边距为20px, 左右自动 */
8.      max-width: 800px; /* 设置最大宽度 */
9.      padding: 20px; /* 设置内边距 */
10.     background-color: #fff; /* 设置背景颜色 */
11.     border-radius: 10px; /* 设置圆角 */
12.     box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1); /* 添加阴影 */
13.     text-align: center; /* 设置文本居中 */
14. }
15.
16. .lyrics p {
17.     margin: 10px 0; /* 设置段落的上下外边距 */
18. }

```

明月几时有？把酒问青天。
 不知天上宫阙，今夕是何年。
 我欲乘风归去，又恐琼楼玉宇，高处不胜寒，起舞弄清影，
 何似在人间。
 转朱阁，抵绮户，照无眠。
 不应有恨，何事偏向别时圆。
 人有悲欢离合，月有阴晴圆缺，此事古难全。
 但愿人长久，千里共婵娟。

音视频部分 html 代码:

```

1.  <h2>音频播放</h2>
2.      <audio controls>
3.          <source src="resources/embed/蔡琴明月几时有.mp3" type="audio/mpeg">
4.          您的浏览器不支持音频元素。
5.      </audio>
6.
7.  <h2>视频播放</h2>
8.      <video controls width="600">
9.          <source src="resources/puppy.mp4" type="video/mp4">
10.         您的浏览器不支持视频元素。
11.      </video>

```


<audio controls>: HTML5 的音频元素, controls 属性表示显示默认的音频控制界面, 包括播放、暂停、音量调节等按钮。

关于备用内容: **您的浏览器不支持视频元素**

浏览器首先尝试解析和加载 <audio> 标签内的媒体文件, 如果浏览器支持 <audio> 标签并且能够播放指定的音频格式(如 MP3), 则会显示音频播放器, 用户可以正常播放音频, 如果浏览器不支持 <audio> 标签, 或者不支持指定的音频格式, 那么它会忽略无法识别的部分, 转而显示 <audio> 标签内的备用内容, 即 您的浏览器不支持音频元素。

备用内容的考虑因素:

用户友好性: 可以告知用户故障的原因, 避免浏览器页面空白的错误, 提升了用户的体验。

兼容性: 考虑到浏览器版本不兼容, 确保在不同的环境显示合理的信息。

结果展示:



项目 3 设计简易网站导航

代码解释：使用段落标记和超链接实现导航：

```
1. <h2>使用段落标记和超链接实现导航</h2>
2. <header>
3.   <nav>
4.     <p>
5.       <a href="http://www.baidu.com/">百度</a> |
6.       <a href="http://www.sina.com.cn/">新浪</a> |
7.       <a href="http://www.qq.com/">腾讯</a> |
8.       <a href="http://www.sohu.com/">搜狐</a> |
9.       <a href="http://www.163.com/">网易</a> |
10.      <a href="http://www.google.com.hk/">谷歌</a>
11.    </p>
12.  </nav>
13. </header>
```

解释：

实现效果：

网页顶部会显示一个简单的文本导航栏，包含六个链接。

用户可以点击这些链接，快速访问相应的网站。

链接之间的竖线使导航栏整洁有序。

这种方法适用于简单的导航，对于更复杂的导航，通常会使用无序列表。

使用无序列表实现导航：

```
1. <h2>使用无序列表实现导航</h2>
2. <footer>
3.   <nav>
4.     <ul class="nav-list">
5.       <li><a href="http://www.baidu.com/">百度</a></li>
6.       <li><a href="http://www.sina.com.cn/">新浪</a></li>
7.       <li><a href="http://www.qq.com/">腾讯</a></li>
8.       <li><a href="http://www.sohu.com/">搜狐</a></li>
9.       <li><a href="http://www.163.com/">网易</a></li>
10.      <li><a href="http://www.google.com.hk/">谷歌</a></li>
11.    </ul>
12.  </nav>
13. </footer>
```

解释：

1. `<h2>`使用无序列表实现导航`</h2>`：标题，描述导航的实现方式。
2. `<footer>`：定义文档的底部区域。
3. `<nav>`：定义导航链接的区域。
4. `<ul class="nav-list">`：使用无序列表 `` 包含多个列表项 ``，每个列表项包含一个超链接 `<a>`，指向不同的外部网站。

使用无序列表来实现导航有以下几个好处：

1. **语义清晰**：无序列表自然地表达了一组项目的集合，导航菜单本身就是一系列链接的集合。使用 `` 和 `` 标签可以使 HTML 结构更符合内容的逻辑，增强网页的语义化，有利于搜索引擎理解页面结构，提高 SEO 效果。
2. **样式灵活**：列表结构方便使用 CSS 进行样式定制。通过为 `` 和 `` 设置类名（如 `class="nav-list"`），可以轻松地控制导航菜单的布局 and 外观，例如横向排列、垂直排列、添加间距、设置背景和鼠标悬停效果等。
3. **可访问性提升**：使用语义化的列表结构，辅助技术（如屏幕阅读器）可以更准确地解析和朗读导航菜单，提升对残障用户的友好度。
4. **易于维护**：添加或删除导航项时，只需增减 `` 元素，结构清晰，修改方便。这使得导航菜单的更新和维护更加快捷。
5. **结构清晰**：列表结构使 HTML 代码更有条理，便于开发者阅读和理解，减少了代码混乱的可能性。

结果展示：



项目 4 设计新生课程简介

重要方法:

页面内导航

实现方法:

1. 创建导航链接:

在页面的导航部分, 使用 `<a>` 标签创建链接, `href` 属性设置为目标元素的 `id`, 前面加上 `#` 符号。

在页面的导航部分, 使用 `<a>` 标签创建链接, `href` 属性设置为目标元素的 `id`, 前面加上 `#` 符号。如下:

```
<ul>
  <li><a href="#english-courses">英语课程</a></li>
  <li><a href="#math-course">高等数学</a></li>
  <li><a href="#physics-course">大学物理</a></li>
</ul>
```

这里的 `href="#english-courses"` 表示当用户点击“英语课程”链接时, 页面会跳转到 `id` 为 `english-courses` 的元素位置。

2 设置目标位置:

在页面的相应位置, 为目标元素添加 `id` 属性, 使其与导航链接中的 `href` 值匹配。

```
<h2 id="english-courses">英语课程</h2>
<p>这里是关于英语课程的详细介绍。</p>

<h2 id="math-course">高等数学</h2>
<p>这里是关于高等数学的详细介绍。</p>

<h2 id="physics-course">大学物理</h2>
<p>这里是关于大学物理的详细介绍。</p>
```

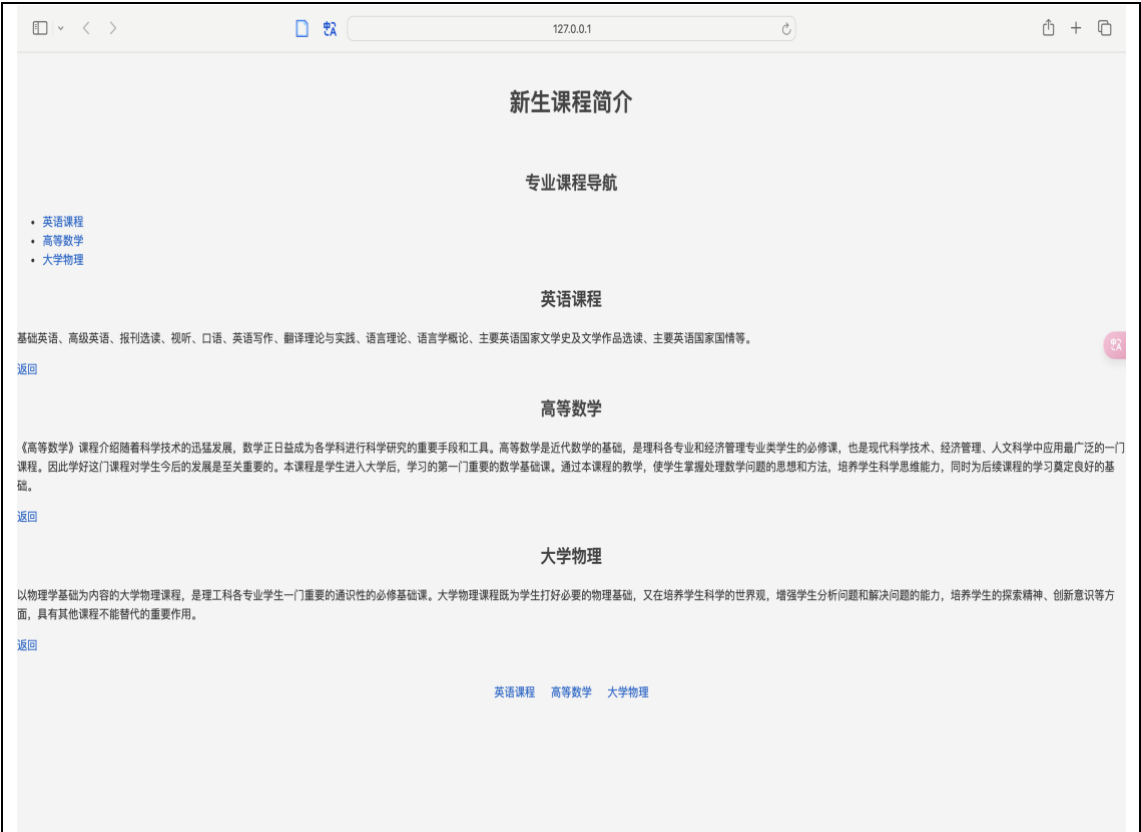
工作原理:

- 当用户点击导航链接时, 浏览器会在当前页面查找匹配的 `id`, 然后自动滚动到该元素的位置。
- `#` 符号用于指定页面内的锚点链接, 后面的字符串必须与目标元素的 `id` 完全一致。

优点:

- **提升用户体验:** 用户可以快速导航到感兴趣的内容, 无需手动滚动页面。
- **实现简单:** 只需使用标准的 `HTML` 标签, 无需额外的脚本或插件。
- **兼容性好:** 所有现代浏览器都支持这种页面内跳转方式。

结果展示:



课外拓展训练

1.

程序介绍：

对整个展览应用 gallery 样式：

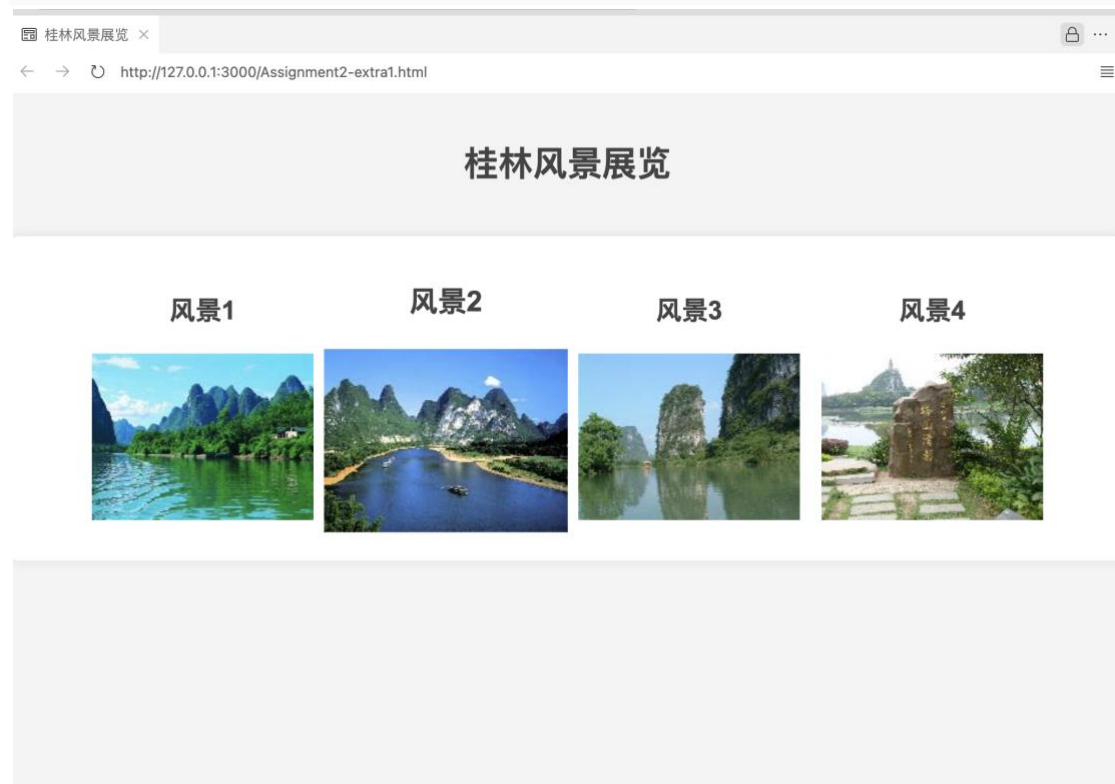
同样使用 flex 布局

/ 画廊样式 */*

```
.gallery {  
  display: flex; /* 使用 Flexbox 布局 */  
  justify-content: center; /* 子元素水平居中 */  
  flex-wrap: wrap; /* 子元素在必要时换行 */  
}
```

对单张图片应用 css 样式实现悬停缩放等功能

```
/* 缩略图样式 */  
.thumbnail {  
    cursor: pointer; /* 鼠标悬停时显示为指针 */  
    transition: transform 0.2s; /* 添加缩放过渡效果 */  
    margin: 10px; /* 设置缩略图的外边距 */  
}  
  
.thumbnail:hover {  
    transform: scale(1.1); /* 鼠标悬停时缩放 10% */  
}
```



2.中国名牌大学导航

```
/* 大学链接容器样式 */  
.university-links {  
    display: flex; /* 使用 Flexbox 布局 */  
    justify-content: center; /* 子元素水平居中 */  
    flex-wrap: wrap; /* 子元素在必要时换行 */  
    gap: 20px; /* 设置子元素之间的间距 */  
    padding: 20px; /* 设置容器的内边距 */  
    background-color: #fff; /* 设置背景颜色 */  
    border-radius: 12px; /* 设置圆角 */  
    box-shadow: 0 4px 12px rgba(0, 0, 0, 0.1); /* 添加阴影 */  
}
```

```
max-width: 800px; /* 设置最大宽度 */
margin: 20px auto; /* 设置上下外边距为20px，左右自动 */
text-align: center; /* 设置文本居中 */
}
```

以下为针对超链接的 css 样式：

```
/* 超链接样式 */
.university-links a {
  color: #0066cc; /* 设置链接颜色 */
  text-decoration: none; /* 去除链接下划线 */
  padding: 10px 20px; /* 设置内边距 */
  background-color: #e6f7ff; /* 设置背景颜色 */
  border-radius: 8px; /* 设置圆角 */
  transition: background-color 0.3s ease; /* 添加背景颜色过渡效果 */
}

.university-links a:hover {
  background-color: #cceeff; /* 鼠标悬停时改变背景颜色 */
  text-decoration: underline; /* 鼠标悬停时显示下划线 */
}
```



因为使用了 **flex** 布局，所以超链接文本块可以灵活美观的展示在任何比例上。

知识总结：

灯箱画廊的实现：

HTML 结构： 使用<div>和标签创建图片缩略图。

CSS 样式： 利用 Flexbox 布局，使元素居中对齐；使用 `position: fixed;` 和 `z-index` 控制灯箱的显示层级。

JavaScript 交互： 通过事件监听，实现点击缩略图弹出大图，点击灯箱区域关闭大图的功能。

媒体元素的嵌入：

音频和视频标签： 使用<audio>和<video>标签在网页中添加音频和视频内容。

备用内容： 提供备用文本，提示用户浏览器不支持相关媒体元素，提高了网页的兼容性和可访问性。

导航栏的创建：

段落标签与超链接： 使用<p>标签和<a>标签创建简单的导航栏。

无序列表的优势： 使用和标签构建导航菜单，语义清晰，便于样式定制和维护。

页面内跳转与返回顶部功能：

锚点链接： 使用 `href="#id"` 和对应的 `id` 属性，实现页面内的快速跳转。

返回顶部： 添加返回链接，并在页面顶部设置对应的锚点，实现返回顶部的功能。

平滑滚动效果： 使用 CSS 的 `scroll-behavior: smooth;` 或 JavaScript 的 `window.scrollTo` 方法，增强用户体验。

CSS 选择器与样式应用：

如 `.university-links a`，用于选中特定区域内的元素，精确控制样式的作用范围。
使用 `:hover` 和 `transition`，实现鼠标悬停时的样式变化和动画效果。

网页结构的语义化：

语义化标签： 使用 `<header>`、`<nav>`、`<section>`、`<footer>` 等标签，提高网页的可读性和 SEO 性能。

结构清晰： 通过合理的 HTML 标签和属性，使网页结构更清晰，代码更易于维护。
增强用户体验的技巧：

交互性： 通过 JavaScript 实现动态效果，提高网页的交互性。

样式美化： 使用 CSS 进行样式美化，如设置背景色、圆角、阴影等，使页面更美观。

兼容性与可访问性：

提供多种媒体格式和备用内容，确保在不同浏览器和设备上都能正常使用。

指导教师批阅意见：

成绩评定： 分

指导教师签字：

年 月 日

备注: