

华中科技大学

课程实验报告

课程名称： 新生实践课

专业班级 计算机类 2405

学 号 U202414657

姓 名 甘卓锐

指导教师 陈加忠

报告日期 2024 年 11 月 17 日

计算机科学与技术学院

目 录

1	网页整体框架	1
2	分页面设计	2
2.1	主页设计	2
2.2	About Me	3
2.3	Educational Background	3
2.4	Links	3
2.5	Nav / Footer	4
3	网页设计小结	5
4	课程的收获和建议	6
4.1	建议	6
4.2	L ^A T _E X	6
4.3	Python	6
4.4	Git	6
4.5	网页制作	6
A	附录 TypeScript 代码	7
A.1	Hello World!	7
A.2	About Me	11
A.3	Educational Background	12
A.4	Links	14
A.5	Nav / Footer	15

1 网页整体框架

我的个人主页框架如图1-1。网址是<https://galliumnitride-dl.github.io/>。

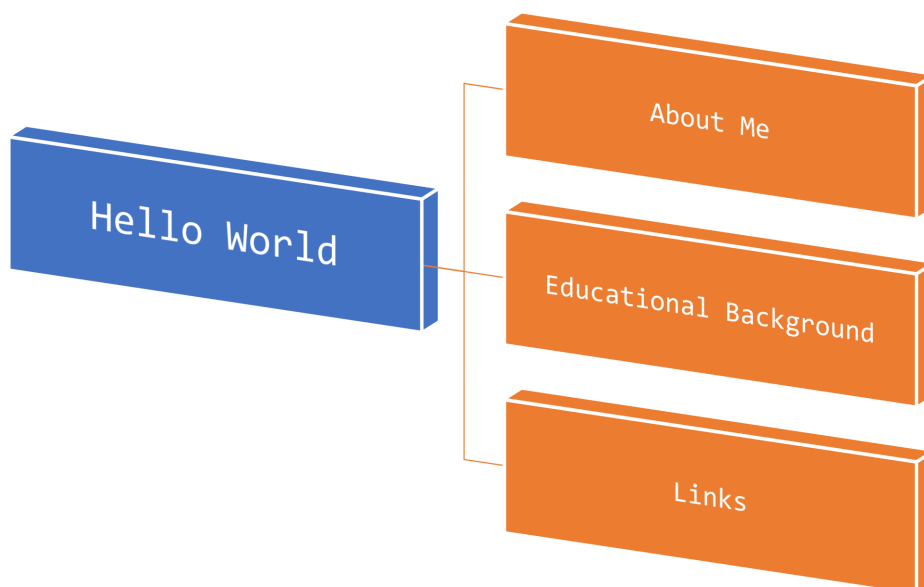


图 1-1 个人网页框架

通过一个主页 *Hello World*，链接到三个分页面：*About Me*、*Educational Background*、*Links*。三个页面分别介绍了我的个人信息、教育背景和社交平台链接。每个页面上方都有一个导航栏，可以方便地切换到其他页面。

2 分页面设计

在 TypeScript 的帮助下，可以创建更加复杂的网页效果。

2.1 主页设计

在主页中，使用 TypeScript 的异步方法，实现了循环变更的文字，如图2-1所示。

```
frontPageTextAnimation().then(() => setTimeout(frontPageTextAnimation, callInterval));

async function frontPageTextAnimation() {
  let animInterval : number = 75;
  let originalText : string = animText.textContent!;
  let i : number = originalText.length;
  while (i) {
    animText.textContent = originalText.substr(0, i--);
    await new Promise(r => { (value: unknown) => void => setTimeout(r, animInterval); });
  }

  let text : string = texts[currentTextIndex].text;
  animText.style.color = texts[currentTextIndex].color;
  let j : number = 0;
  while (j <= text.length) {
    animText.textContent = text.substr(0, j++);
    await new Promise(r => { (value: unknown) => void => setTimeout(r, animInterval); });
  }
  currentTextIndex = (currentTextIndex + 1) % texts.length;
}
```

图 2-1 循环文字代码部分

通过 TypeScript 对网页中元素的便捷获取和修改，实现了主页正文文字变化的自动初始化，避免了繁杂的操作。如2-2的 HTML 即可实现2-3的效果。

```
<div id="link-to-lnk">
  <h2>Links</h2>
  <p>Links to my social media and other platforms.</p>
  
  <a href="links.html"></a>
</div>
```

图 2-2 HTML 元素

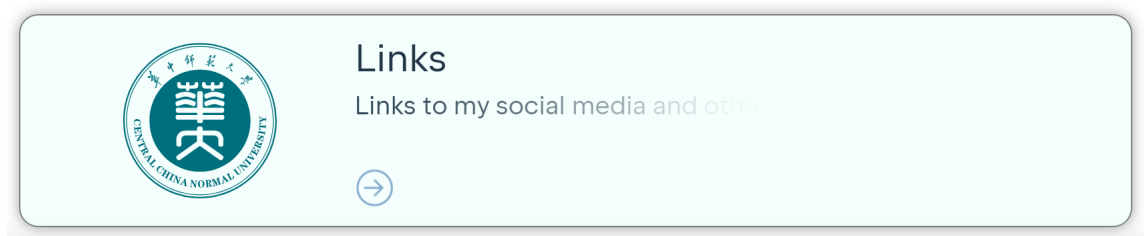


图 2-3 实现的效果

同时，将 CSS 中的 *height: 100vh* 属性应用到 *div* 元素上，实现了窗口整页的

填充，并用 *box-shadow* 加入了阴影效果，使页面更加美观。

2.2 About Me

在 *About Me* 页面中，通过对鼠标事件的监听和 CSS 中的 *scroll-snap-align* 属性，实现了类似幻灯片放映的效果，如图2-4所示。



图 2-4 滚动的页面

2.3 Educational Background

在 *Educational Background* 页面中，通过按钮调用 TypeScript 方法，实现在点击左侧对应的按钮时更改右侧的文字说明，利用 CSS 的 *height: max-content*，实现了背景边框的自动缩放，如图2-5所示。

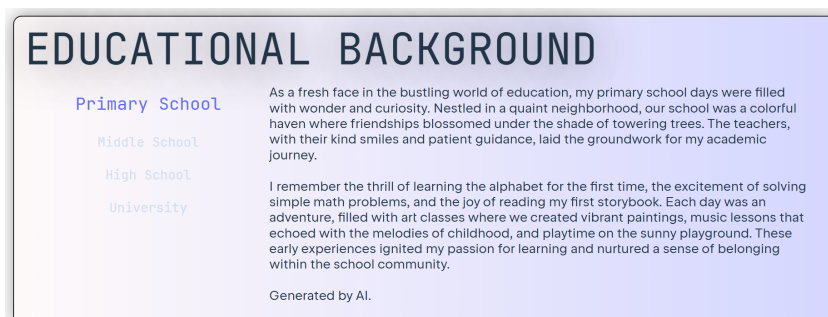


图 2-5 效果

2.4 Links

在 *Links* 页面中，通过 CSS 中的 *transition* 和 *hover* 功能，实现了鼠标悬停时的放大和变色效果，如图2-6所示。

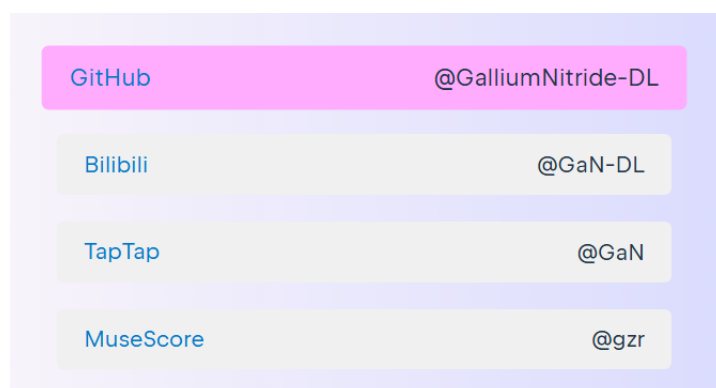


图 2-6 鼠标悬停效果

2.5 Nav / Footer

在每个页面的顶部和底部，都有一个导航栏和页脚。在网站加载时，利用 TypeScript 自动加载导航栏和页脚，只需要在 HTML 中添加 `<nav>` 和 `<footer>` 标签并在 TypeScript 中调用相应的方法即可，如图2-7和2-8所示。

```
<body>

  <nav></nav>

  <div class="container">

    <footer></footer>

  </div>

</body>
```

图 2-7 HTML 元素



图 2-8 效果

3 网页设计小结

CSS 是网页设计的重要组成部分，通过 CSS 的属性设置，可以实现网页的美化和动态效果。但是，CSS 提供的属性多而繁杂，不易掌握。但是只要了解了常用属性和布局，前端设计总体来说还是比其它的程序设计更加直接易懂。

TypeScript 和 JavaScript 是使网页拥有更复杂的功能和效果的有效工具。通过这些代码可以实现动态修改网页元素、监听事件、实现动画效果等。在今后的 H5 设计中，它们的应用也更加广泛。

4 课程的收获和建议

4.1 建议

无。

4.2 L^AT_EX

学习了 L^AT_EX 排版工具，可以用于生成高质量的文档。在学习中发现，L^AT_EX 的排版效果非常好，但是入门难度较高，环境配置复杂，学习成本较高。但是从现成的例子入手可以简化上手难度。

4.3 Python

强大的高级编程语言，可以用于数据处理、人工智能等多个领域。在学习中发现，Python 的很多概念与 C 语言十分相似，但是 Python 的风格更加直白易懂，与 C 语言相比，入门难度更低。

4.4 Git

版本管理工具。在多人合作项目的时候会用到。

4.5 网页制作

WebStorm 是非常好的 IDE。在学习的过程中，自主研究了 Node.js，以及 Vite，这些工具可以使网页的功能更加丰富、制作更加便捷。除此之外，了解到了前端还有许多常用框架，如 *Vue*、*React* 等，它们可以使网页模块化、组件化，提高开发效率。我希望在今后的实践项目中，可以尝试使用这些框架。

A 附录 TypeScript 代码

Vite + TS 在构建时会将 TS 代码进行编译。下面给出编译前的完整 TS 代码。

A.1 Hello World!

indexpage.ts

```
import './indexpage.css';
import { setupParagraphComponent } from './ParagraphComponent.ts';
import { initNavAndFooter } from '../globalStuff.ts';

document.addEventListener('DOMContentLoaded', () => {
  initNavAndFooter();
  setupParagraphComponent('link-to-about-me');
  setupParagraphComponent('link-to-edu');
  setupParagraphComponent('link-to-lnk');

  let callInterval = 4500;
  const animText =
document.getElementById('frontpageTextAnimation')!;
  let currentTextIndex = 0;
  let texts = [
    { text: "coding", color: "gray" },
    { text: "to have fun", color: "blue" },
    { text: "his friends", color: "purple" },
    { text: "playing", color: "green" },
    { text: "enjoying life", color: "brown" },
    { text: "this world", color: "red" },
    { text: "himself", color: "orange" },
    { text: "HUST", color: "magenta" }
```

```
];

frontPageTextAnimation().then(() =>
setInterval(frontPageTextAnimation, callInterval));

async function frontPageTextAnimation() {
  let animInterval = 75;
  let originalText = animText.textContent!;
  let i = originalText.length;
  while (i) {
    animText.textContent = originalText.substring(0, i--);
    await new Promise(r => setTimeout(r, animInterval));
  }

  let text = texts[currentTextIndex].text;
  animText.style.color = texts[currentTextIndex].color;
  let j = 0;
  while (j <= text.length) {
    animText.textContent = text.substring(0, j++);
    await new Promise(r => setTimeout(r, animInterval));
  }
  currentTextIndex = (currentTextIndex + 1) % texts.length;
}
});
```

ParagraphComponent.ts

```
import arrow from '../img/Picture2.png'

export function setupParagraphComponent(elementID: string)
```

```
{
  let div = document.getElementById(elementID)!;
  let title = div.querySelector('h2')!.textContent || 'No Title';
  let text = div.querySelector('p')!.textContent || 'No Text';
  let img = div.querySelector('img')!.src || '';
  let href = div.querySelector('a')!.href || '';
  div.innerHTML =
    `
    <div class='paragraph-component'>
      
      <div class="card-content">
        <h2><span>${title}</span></h2>
        <p>${paragraphAnimationSpan(text, 0.5)}</p>
        <a href="${href}">
          
        </a>
      </div>
    </div>
    `

  const observer = new IntersectionObserver((entries) =>
  {
    let isAnimationActivated = false;
    entries.forEach((entry) =>
    {
      if (entry.isIntersecting && !isAnimationActivated)
      {
        const imgs = entry.target.querySelectorAll('img');
        const spans =
```

```
entry.target.querySelectorAll('span');
    const paragraphComponent =
entry.target.querySelector('.paragraph-component')!;

    paragraphComponent.classList.add
('background-fade-in');
    imgs.forEach((img) =>
    {
        img.classList.add('translate-in-from-left');
    });
    spans.forEach((span) =>
    {
        span.classList.add('fade-in');
    });
    observer.disconnect();
    isAnimationActivated = true;
    }
    });
    }, { threshold: 0.8 });

    observer.observe(div);
}

function paragraphAnimationSpan(text: string, initialDelay = 0.5,
perCharDuration = 0.01)
{
    return text.split('').map((char, i) =>
        `
```

```
}
```

A.2 About Me

aboutme.ts

```
import './aboutme.css';
import './globalStyle.css'

import { initNavAndFooter } from '../globalStuff.ts';
import { initSlideShow } from './SlideShowLogic.ts';

initNavAndFooter();

document.addEventListener('DOMContentLoaded', () => {
    initSlideShow();
});
```

SlideShowLogic.ts

```
export function initSlideShow()
{
    const sections = document.querySelectorAll('section');
    let currentSectionIndex = 0;

    let isScrolling = false;

    document.addEventListener('wheel', (event) => {
        if (isScrolling) return;
```

```
        if (event.deltaY > 0 && currentSectionIndex <
sections.length - 1) {
            currentSectionIndex++;
        } else if (event.deltaY < 0 && currentSectionIndex > 0) {
            currentSectionIndex--;
        }

        isScrolling = true;
        sections[currentSectionIndex].scrollIntoView({ behavior:
'smooth' });

        setTimeout(() => {
            isScrolling = false;
        }, 1500);
    });
}
```

A.3 Educational Background

education.ts

```
import './education.css';
import { initNavAndFooter } from '../globalStuff.ts';
import { initSelection } from './SectionSelectHandler.ts';

document.addEventListener('DOMContentLoaded', () => {
    initNavAndFooter();
    initSelection();
});
```

SectionSelectHandler.ts

```
let currentSelection = 0;
let texts: string[];
let selections: Element[];

export function initSelection()
{
    selections = Array.from(document.querySelectorAll
('#about-me-left button'));
    selections.forEach((button, index) =>
    {
        button.classList.add('deselected');
        button.addEventListener('click', () => select(index));
    });

    var textElements = document.querySelectorAll
('#about-me-right h4');
    texts = Array.from(textElements, element =>
element.textContent || '');

    if (selections.length !== texts.length)
        throw new Error('The number of selections and texts are
not equal');

    currentSelection = 0;
    selections[currentSelection].classList.remove('deselected');
    selections[currentSelection].classList.add('selected');
    updateText();
}
```

```
export function select(index: number)
{
    if (index === currentSelection) return;
    selections[currentSelection].classList.remove('selected');
    selections[currentSelection].classList.add('deselected');
    selections[index].classList.remove('deselected');
    selections[index].classList.add('selected');
    currentSelection = index;
    updateText();
}

function updateText()
{
    document.querySelector('#about-me-right')!.innerHTML = `
    <h4>
        ${texts[currentSelection].replace(/\n/g, '<br>')}
    </h4>
    `;
}
```

A.4 Links

links.ts

```
import './links.css'
import '../globalStyle.css';

import { initNavAndFooter } from '../globalStuff.ts';
```



```
document.addEventListener('DOMContentLoaded', () => {  
    initNavAndFooter();  
});
```

A.5 Nav / Footer

globalStuff.ts

```
// Initializes the Nav and footer  
import './nav.css'  
import './globalStyle.css'  
  
import { setupNavigationBarAnimation } from  
"./NavigationBarAnimation.ts";  
  
export function initNavAndFooter()  
{  
    document.querySelector('nav')!.innerHTML = `  
        <span class="nav-label">GaN's WEBPAGE</span>  
        <div class="nav-button">  
            <a href="/index.html">Hello World!</a>|  
            <a href="/aboutme.html">About Me</a>|  
            <a href="/education.html">Educational  
Background</a>|  
            <a href="/links.html">Links</a>  
        </div>  
    `;  
    setupNavigationBarAnimation();  
  
    document.querySelector('footer')!.innerHTML = `
```

```
<h6>2024.11 | GaN's webpage.</h6>

`;  
}
```

NavigationBarAnimation.ts

```
let lastScrollY = 0;  
const nav = document.querySelector('nav')!;  
  
export function setupNavigationBarAnimation()  
{  
  window.addEventListener('scroll', () =>  
  {  
    let scrollY = window.scrollY;  
    if (scrollY > lastScrollY)  
    {  
      nav.classList.remove('nav-in', 'nav-out');  
      nav.classList.add('nav-out');  
    } else  
    {  
      nav.classList.add('nav-in', 'nav-out');  
      nav.classList.remove('nav-out');  
    }  
    lastScrollY = scrollY;  
  });  
}
```