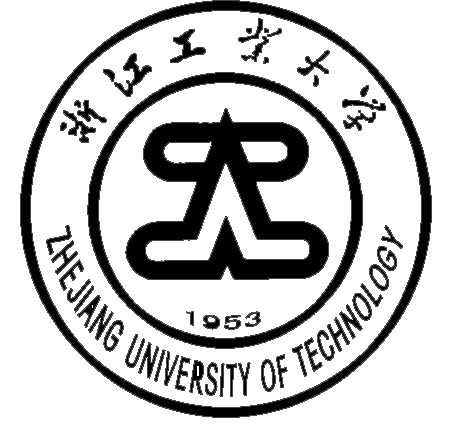


Web前端开发

期末报告



题目 Dilidili设计与实现

姓名 苏怡鹏

学号 202003151119

班级 软工大数据2001

任课老师 简琤峰

提交日期 2022.6.5

目录

[一、 网站项目说明 2](#_Toc105334157)

[二、 网站各项目详细说明 2](#_Toc105334158)

[1. 网站内容完整性 2](#_Toc105334159)

[1.1网页完整性 2](#_Toc105334160)

[1.2 主页完整性 5](#_Toc105334161)

[1.3 网页元素多样性 7](#_Toc105334162)

[2. 网站样式设计说明 9](#_Toc105334163)

[2.1整体设计 9](#_Toc105334164)

[2.2 元素样式设计 10](#_Toc105334165)

[3. 网站交互功能 12](#_Toc105334166)

[3.1 数据验证 12](#_Toc105334167)

[3.2 用户状态 12](#_Toc105334168)

[3.3 图片切换 13](#_Toc105334169)

[3.4 图形绘制 13](#_Toc105334170)

[3.5 浮动 13](#_Toc105334171)

[3.6 拖放 14](#_Toc105334172)

[3.7分页导航 14](#_Toc105334173)

[4. 特色 15](#_Toc105334174)

[4.1 设计亮点 15](#_Toc105334175)

[4.2技术亮点 15](#_Toc105334176)

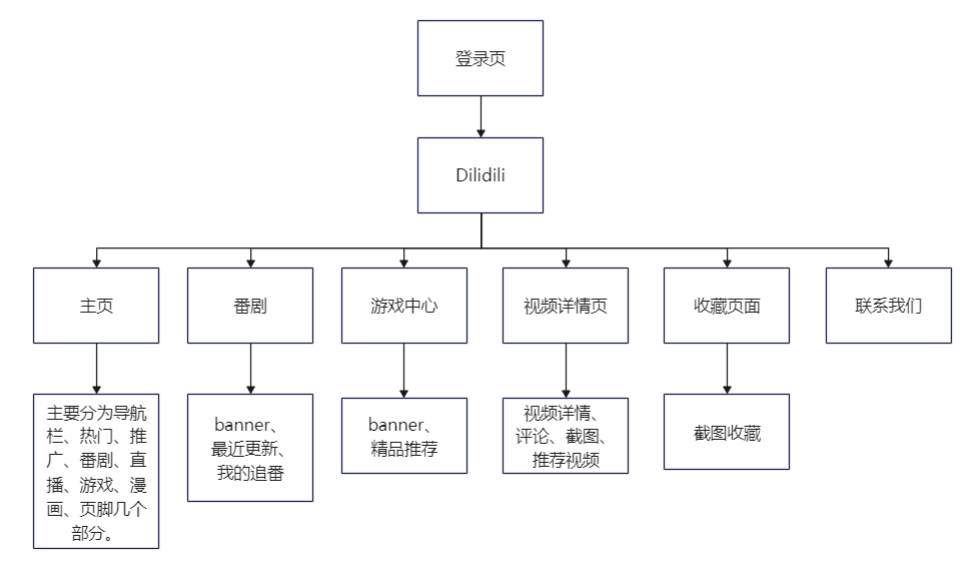
[三、 个人总结体会 17](#_Toc105334177)

## 网站项目说明

Dilidili是一个ACG网站，上面有各种动画、漫画、游戏等资源，用户可以在这里结识同好、找到许多欢乐。

网站基于Html5、CSS3和原生JavaScript实现，参照Bilibili的设计，并包含各种常用元素、样式、交互功能，内容丰富，可供前端入门人员参照学习。

网站整体结构如下：



## 网站各项目详细说明

### 网站内容完整性

#### 1.1网页完整性

登录页面：

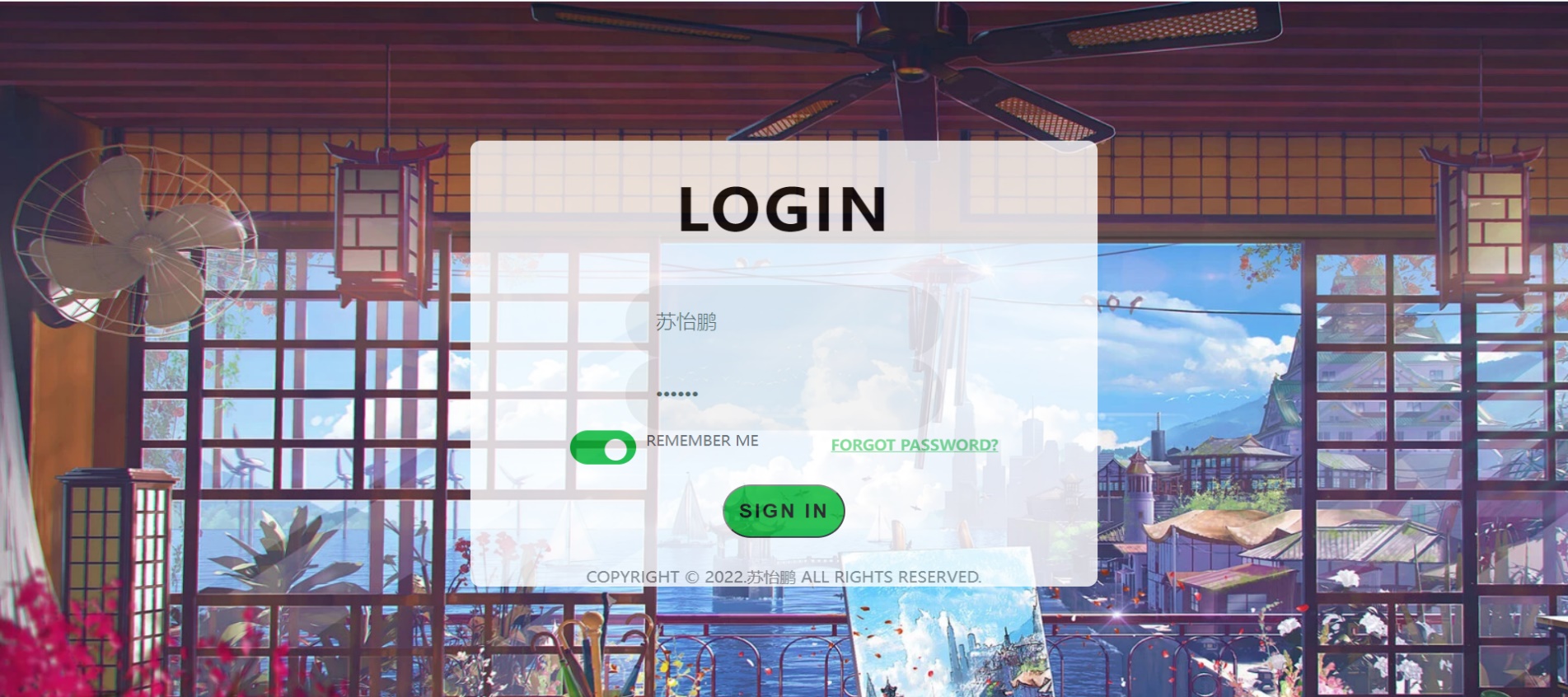


图 1‑1.1

联系页面：



图 1‑1.2

主页面：

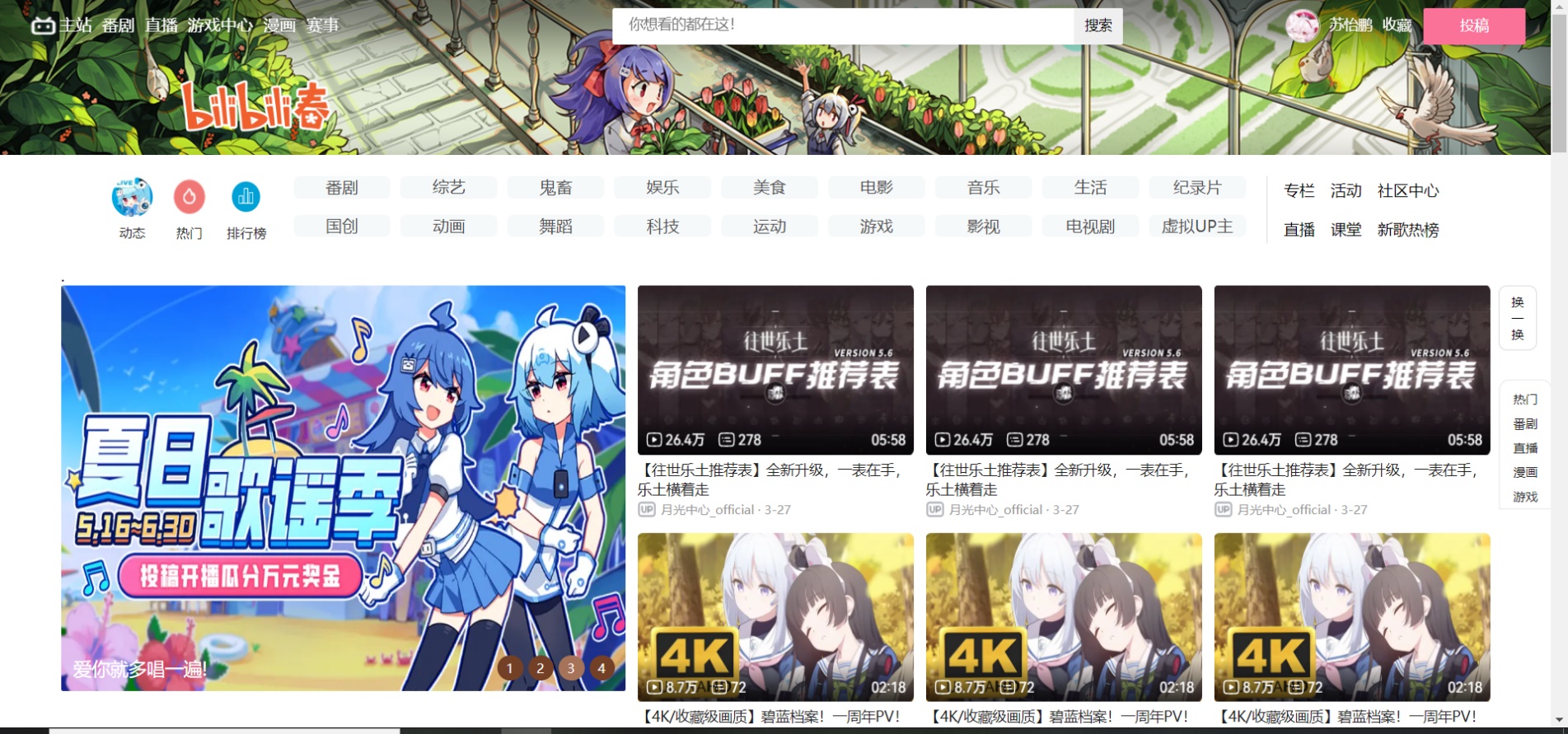


图 1‑1.3 部分主页

分类展示页面-番剧：



图 1‑1.4

分类展示页面-游戏中心：

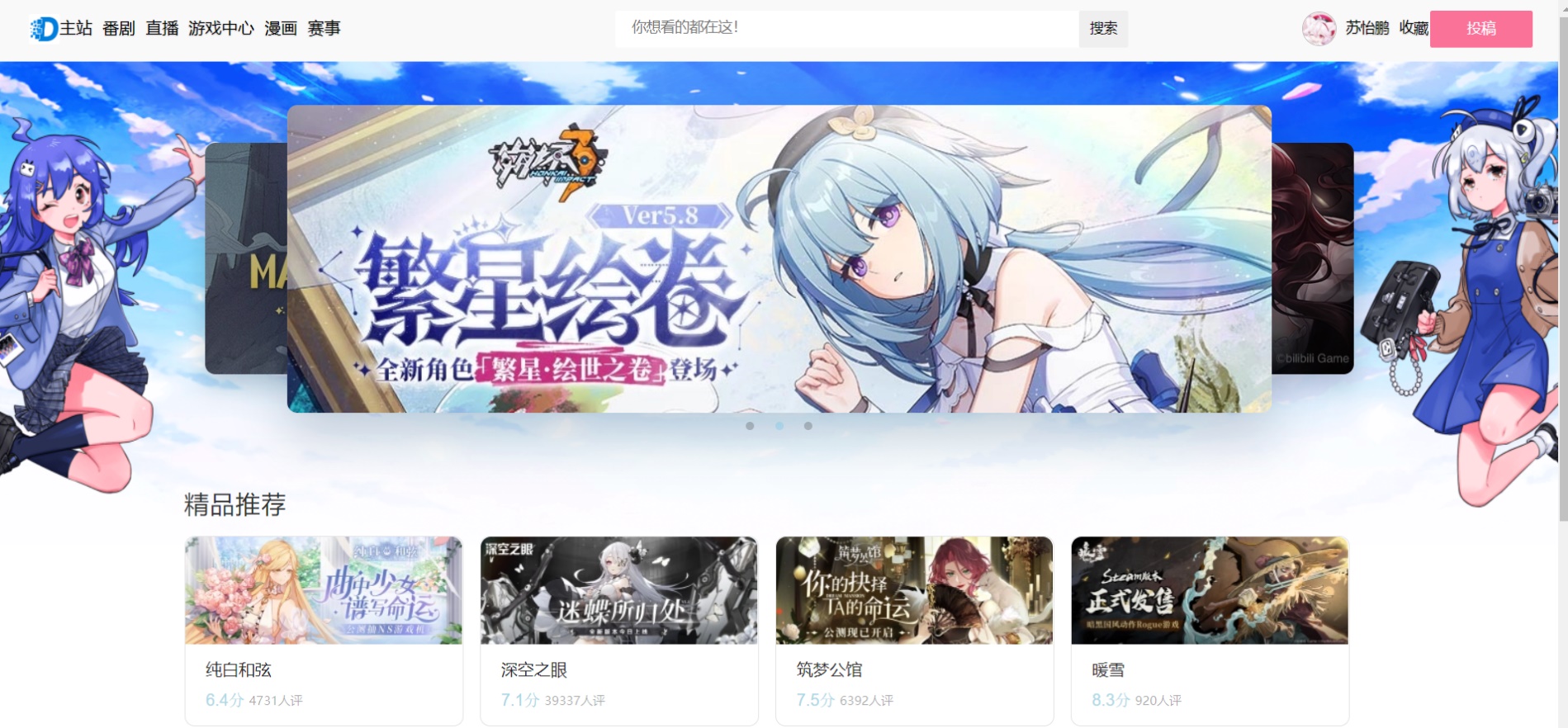


图 1‑1.5

视频详情页：

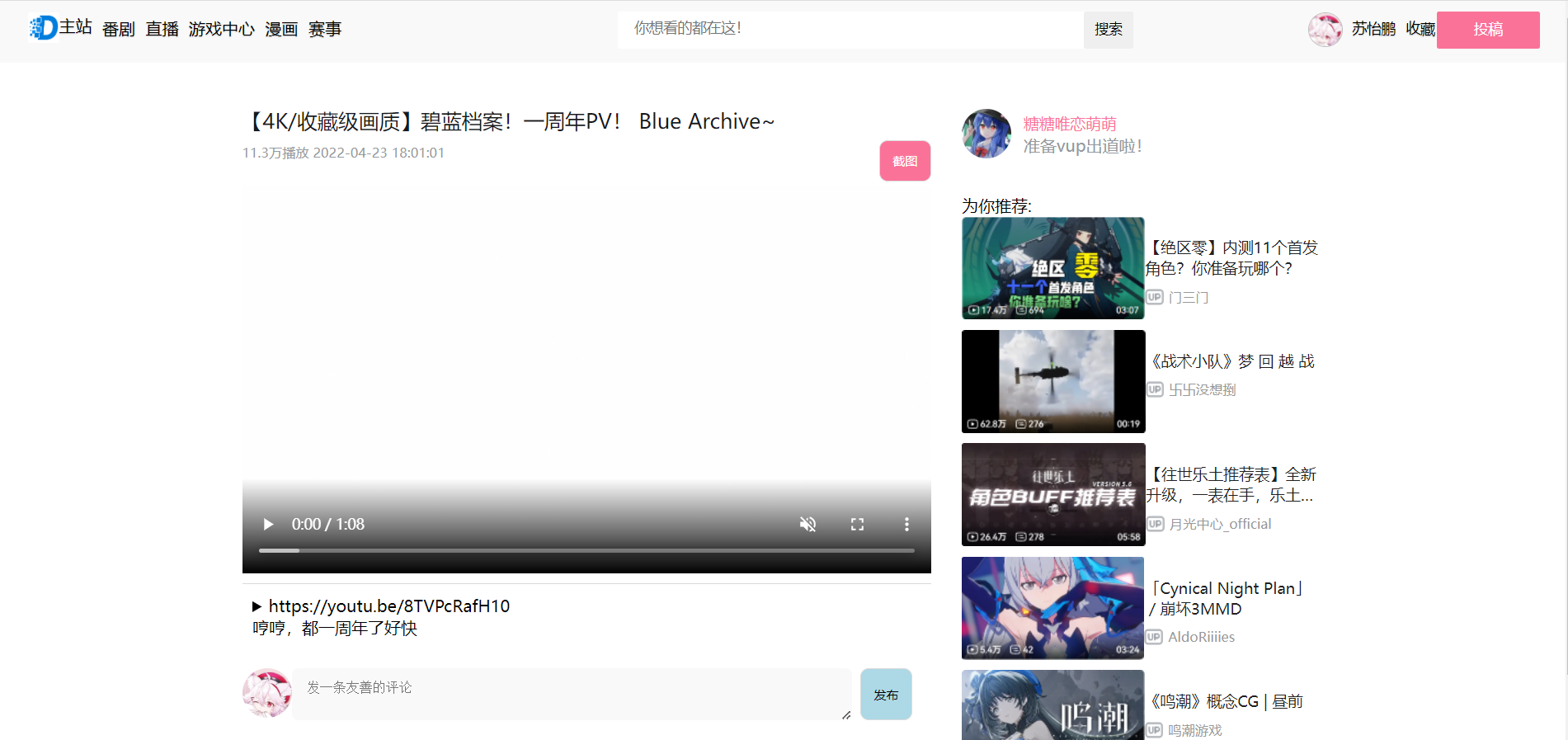


图 1‑1.6

#### 主页完整性

页眉区：



图 1‑1‑2.1

Banner广告区：



图 1‑2.2

正文区：



图 1‑2.3 正文区示例

副栏区：



图 1‑2.4 副栏区示例

页脚区：



图 1‑2.5

#### 网页元素多样性

列表（ul）：



图 1‑1‑3.1 列表示例

图文混排：



图 1‑3.2 图文示例

视频：



图 1‑3.3 视频示例

Canvas：

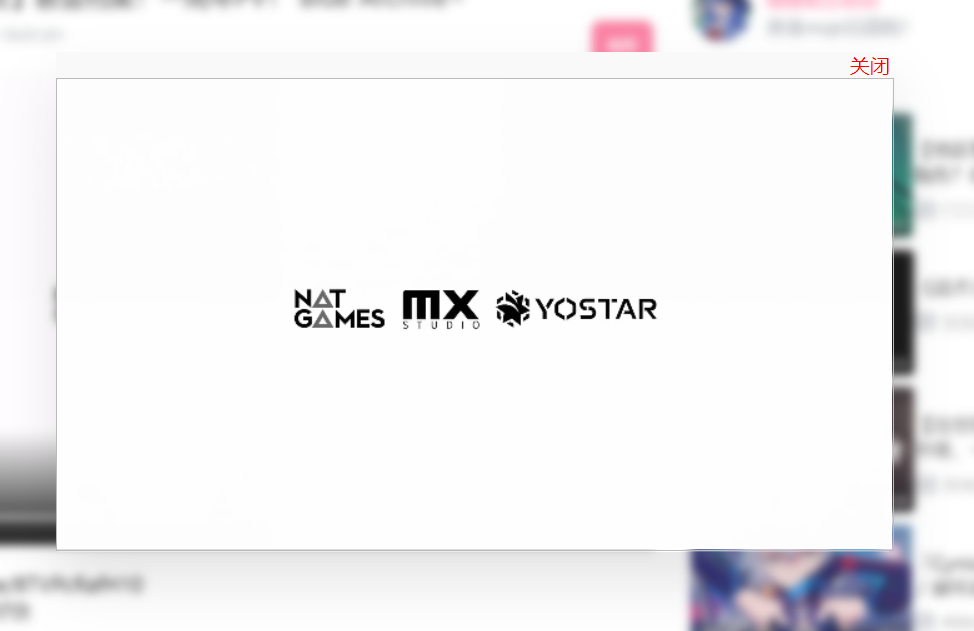


图 1‑3.4 canvas实现截图

### 网站样式设计说明

#### 2.1整体设计

Dilidili整体以白色为主色调，纯净、简约兼容性好，有效突出网站内容。各个分页面布局与边距等设计不尽相同，以下一一介绍。

主页：

主页整体分为三个部分：header、main和footer。

Header部分包括导航栏、banner以及channel三个部分。导航栏整体采用flex布局，左边导航链接部分为横向排列的ul列表，紧接着是搜索框，最右边的用户部分采用绝对位置布局。背景为banner。Channel部分也采用flex布局，外边距与内边距为55px左右，使channel部分整体位于中央，排列紧凑。

Main部分宽1400px，左右内边距56px，包含推荐区、推广区、直播区、番剧区、漫画区、游戏区六大模块。这六个模块均采用grid布局，模块间存在56px的下内边距。具体行列数各分区略有不同。以下以推荐区与直播区为例介绍。

推荐区分为2行5列共10个小网格，其中左边为切换广告，占2行2列4个网格；右边为6个推荐视频，每个视频占1个小网格。直播区整体采用网格布局，分为左右两部分共5列，其中左侧推荐直播占4列，右边排行榜部分占1列。推荐直播内部又分为2行4列共8个小网格，每个推荐主播占1个网格。小网格间上下边距为6px，左右边距为12px。排行榜部分分为上下两块，上方为标题即“排行榜”，下方采用纵向列表布局。列表详细内容在元素样式中说明。其他模块布局与直播区类似，此处不再赘述。

Footer部分较为简单，主要由文字构成，包含联系我们等链接以及版权说明等，文字局中布局。左右外边距为15px，上下内边距为30px。

分页面-番剧：

除导航栏外，页面整体分为上下两块，采用流式布局。上方为banner区，采用justify-content属性将其居中，宽度为整个页面的75%。下方内容区同样采用flex布局，上边距20px与banner区隔开，宽度与其保持一致。内容区又分为左右两部分，均采用float：left布局。

分页面-游戏中心：

页面同样分为上下两块，上方banner区采用默认布局（div block），宽度100%，高度为固定值，无左右边距，详细内容将在后面介绍。下方同样为block布局，宽度为固定值1160px，左右外边距为auto，使其自动居于中央。

视频详情页：

页面分为左右两块，用一个流式布局的div包含，并设置justify-content属性为center使左右两部分相中靠拢，从而无需设置左右边距。左边为视频区域，右边为up主信息与推荐视频，整体都采用默认的block布局，从上而下排列。

联系我们页面：

该页面较为简单，仅包含导航栏、一张背景图片与信息表单，表单使用绝对位置居于页面中央，表单内每行均为block布局，整体文字靠左排列。

收藏页面：

整体分为上下两块，采用flex布局并设置justify-content属性使页面元素居中。上方图片区域采用grid布局，2行2列共4个小网格，每张图片占一个网格。下方分页切换控件采用绝对位置布局，使其位于图片下方、居中排列。

#### 2.2 元素样式设计

导航链接样式设计：

导航栏分为左边分类页面导航链接、中间搜索框和右边用户区三个部分，各个部分均为绝对位置布局，不含背景图片高度60px，浮动于页面最上方。

左边的分类链接区为横向排列的ul列表，采用flex布局，列表元素间左内边距10px。当鼠标悬停在“番剧”和“直播”两个链接上时将出现下拉框。下拉框宽500px，高300px，采用flex布局，分为左右两块，中间用浅灰色线隔开。左边宽350px，内边距20px，采用grid布局，共2列2行，每张图片及其标题占1个网格；右边为列表，包含若干项。

中间搜索框宽446px，高36px，背景为白色，紧接着是浅灰色的搜索button。右边用户区由登录、注册和投稿三个链接构成，内部采用flex布局；登录后则变为用户头像、收藏和投稿三个部分。其中投稿页面尚未实装。

当滚轮位于最上方时，导航栏内文字颜色为白色，背景为透明，与背景图融为一体。滚轮向下移动离开页面最上方时，导航栏内文字颜色变为黑色，背景颜色变为浅灰色，浮于页面最上层。如下图所示：



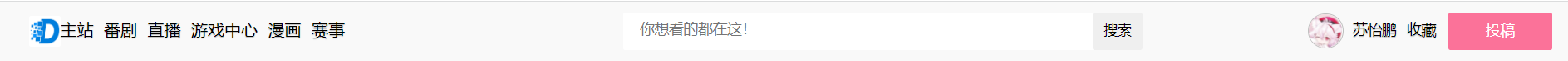


图 2‑2.1 导航栏样式

广告样式设计：

Dilidili共有三种广告样式，此处只介绍第二种。效果如下图所示：



图 2‑2.2 广告样式

整个banner区域采用flex布局，并设置justify-content属性为center，使其居中。Banner宽度为75%，内部使用dl列表，并设置position属性为relative。列表内含dt和dd两部分。

dt部分即为右下角的切换控件，内含4个anchor元素，每个anchor元素由一张小封面构成，宽64px，高48px，右边距5px，边框2px 白色。当鼠标悬停在小封面上方时，其边框颜色变为红色，且自动切换为对应的图片。

dd部分由图片及其左下角的介绍构成。图片铺满整个banner区域，位于底层。左下角介绍部分文字为白色，字体大小18px，为避免图片颜色干扰，对文字的背景设置了由黑到白的渐变色，并设置了背景虚化。

图文混排样式设计：

以如下样式为例介绍：

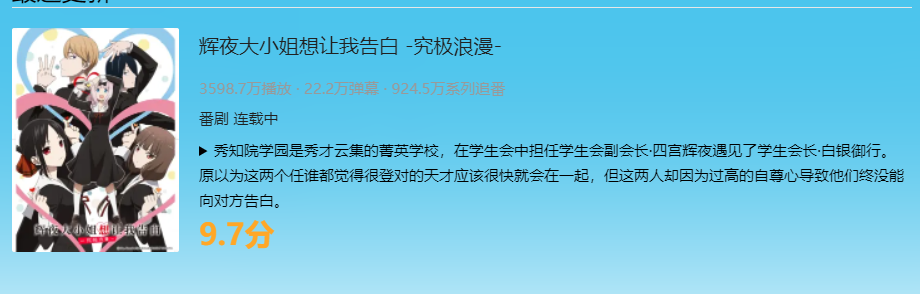


图 2‑2.3 图文混排

区域顶部为高1px的灰色分割线，下方内容区与其间距为16px。内容区分为左右两部分。左边为宽高固定的图片，图片包含于anchor中。为anchor设置向左浮动（float：left）属性，使文字可以居于图片右方。右边文字自上而下分为5个小块。除了简介部分为detail外，其他均包含于div中。各个部分的文字在颜色、大小、字宽、行高、内外边距等方面各有差异，详情不再赘述。

### 网站交互功能

#### 3.1 数据验证

登录页使用JavaScript对输入的密码进行验证，密码为‘123456’时即可登陆成功。

联系我们页面主要通过input标签的pattern属性对姓名和手机号码进行校验。

其中姓名校验方式如下，其含义为2~4个中文字符：

pattern="^[\u4e00-\u9fa5][\u4e00-\u9fa5]{0,2}[\u4e00-\u9fa5]$"

手机号校验方式如下，根据各个运营商的号码区段进行校验，最长11位：

pattern="^(13[0-9]|14[01456879]|15[0-35-9]|16[2567]|17[0-8]|18[0-9]|19[0-35-9])\d{8}$" maxlength=“11”

#### 3.2 用户状态

登录后，导航栏右边用户区发生变化，显示用户的用户名、头像、收藏页链接。如下图所示：

 --> 

登陆成功时，通过js将用户名存入sessionStorage中，跳转到各个页面时，从sessionStorage中读出用户名并将其作为文本替换原来的‘登录’；将‘注册’元素隐藏并显示‘收藏’。

#### 3.3 图片切换

本网站中共有三种图片切换效果，此处介绍较为简单的两种，分别位于主页和番剧页面。

广告主体由一个img元素和一个span元素组成，分别用于显示图片及其标题。另外在右下方设置四个控件，用于图片切换。

首先定义iCount标记当前图片id，定义自动切换函数carousel，通过getElementById获取存放图片的img元素和存放标题的span元素，并将其内图片和标题设置为下一张图片，然后使iCount++。如果iCount>图片数就将其置为1，循环播放图片。通过setInterval函数设置自动切换时间为2s。

然后定义change函数，其功能是根据用户指定的控件切换图片，并更改控件的样式以突出当前播放图片的位置。为每一个空间添加onclick或onmouseover事件，事件内调用change函数，并将iCount修改为对应值。

#### 3.4 图形绘制

利用canvas实现视频截图功能。首先需要为canvas设置width和height属性指定画布的大小。然后为截图按钮添加单击事件：单击后，首先获取视频元素和canvas元素，通过canvas.getContext（‘2d’）函数获取canvas对象，然后用其内置的drawImage函数指定图片来源，并指定绘制图片的位置和宽高，就可以将图片绘制在canvas元素内。

#### 3.5 浮动

为导航栏设置浮动效果。首先需将导航栏设置为fixed布局。然后通过window.onscroll方法为滚动条添加事件：通过id获取导航栏元素。当滚动条移动时，通过document.documentElement.scrollTop方法获取滚动条与页面顶端的距离，当滚动条不处于页面顶端时，将导航栏背景颜色设置为浅灰色，并将导航栏内所有字体颜色修改为黑色，避免文字溶于背景。当滚轮回到页面顶端时，再将导航栏背景修改为transparent，其内所有文字颜色修改回白色。

#### 3.6 拖放

首先为3.4中的canvas元素添加dragstart监听事件，拖放开始时通过事件的dataTransfer对象的setData方法将canvas中数据以‘text/html’形式存入。

然后为导航栏中的‘收藏’元素添加drop监听事件：通过dataTransfer的getData方法获取存入的数据，并将数据以JSON对象的形式存入localStorage中，以便后续收藏页面加载数据。详情将在后面介绍。

此外还需将要拖放的元素的dragable属性设置为true，通过preventDefault方法取消默认的ondragover和ondrop方法。

#### 3.7分页导航

主要实现收藏页面的分页功能。3.6中提到将截图拖放至‘收藏’处即可将截图存入localStorage，此处将所有图片从localStorage中读出，然后按4张图片一页设置若干分页，并按照页面数添加上一页、下一页以及对应若干个分页按钮。定义page变量标记当前分页号。然后使用getElementByName方法获取所有分页按钮（按钮具有相同的名字，其在数组的下标值即为分页号），然后通过IIFE（Immediately Invoked Function Expression，此处用于解决循环内添加事件不能正常执行的问题）为所有按钮添加单击事件：改变按钮边框以突出当前分页号，并通过Switch方法更换分页。更换方法为：将当前分页的display属性设置为none，将其隐藏，然后显示要切换的分页。

显示效果如下：



### 特色

#### 4.1 设计亮点

广告切换效果：

游戏中心页面广告设计如下图所示：



图 4‑1.1

广告由三张图片构成，为每张图片设置不同的z-index值，并分别调整顶层于底层图片的box-shadow与brightness值，然后利用translateX调整三张图片在X轴的偏移量使看起来其层次分明。

自顶向下图片的z-index值分别为989、979、969，当图片切换时，三张图片逆时针转动，顶层图片宽高、亮度变为原来的75%，z-index值变为979，通过调整translateX值使其向左移动，并设置transition属性为0.3s，即在0.3s时间完成切换，从而表现出动态的切换效果。

#### 4.2技术亮点

伸缩盒子：效果如下：



图 4-2.1

当鼠标悬停在区域上时，内部的简介‘弹出’，并设置其边框的阴影效果，表现出伸缩的效果。具体实现为：初始时将简介部分高度设置为0，整个标签设置上边距16px；当鼠标悬停时，将简介部分高度设置为16px，并将上边距调整为0、设置阴影效果，使得交互效果更加明显。

简易评论区：

效果如下：

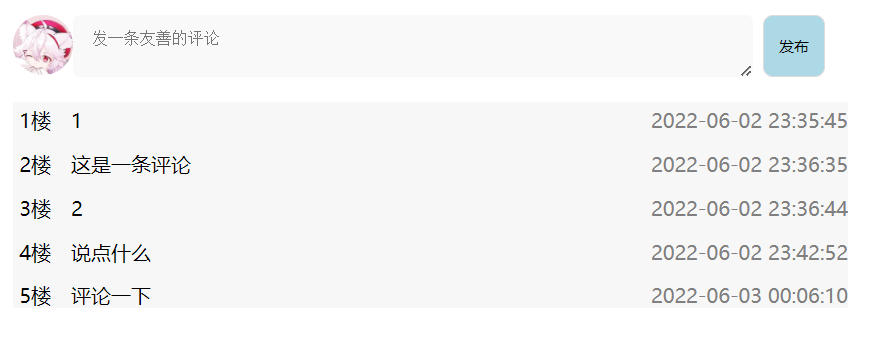


图 4‑2.2

为‘发布’按钮添加单击事件：判断输入框内是否为空，如果为空则用focus方法使输入框成为焦点；否则调用loadStorage方法，将评论内容及评论时间以JSON格式存入localStorage。该方法在页面加载完毕或点击发布后，从localStorage中读出所有评论，然后封装在div元素中，通过innerHtml方法将所有评论加载到评论区。

截图收藏：

前置的截图、拖放、分页功能已经在上面阐述过，此处不再赘述。另外要提的一点是，在canvas拖放到收藏位置时，使用canvas的toDataUrl方法将图片以url形式封装于img标签中，然后与收藏时间一起添加进localStorage的pictures中。这里主要介绍收藏页图片的加载与删除。

定义count变量标记分页数，然后从localStorage中将所有存入的图片读出，按照4张图片一个分页的方式动态设置分页与分页导航按钮（页面主体为网格布局，2行2列），并为每张图片设置一个删除按钮。然后通过getElementByName方法获取按钮数组，并为所有分页导航按钮与删除按钮添加单击事件。当点击删除后，由于删除按钮在数组中的index与图片的index对应，单击后调用splice方法将图片从图片数组中删除，并更新到localStorage中。然后调用reload方法刷新页面，即可实现删除功能。

页面效果如下，详情可打开页面体验：



图 4‑2.3

## 个人总结体会

学习总结;

形象地说，Html是页面的骨架，CSS是页面的皮囊，而JavaScript是页面活的灵魂。

Html由各种标签组成，标签通常成对出现，一个开始一个结束；标签具有标签名、name、id等属性，其中id唯一标志了某个标签。标签内可以有文本，也可以嵌套其他标签。标签间的结构关系构成了一棵DOM树。常用标签有div、span用于划分小区块，a表示超链接，ul、dl等表格，form表单、input输入框等等。还有一些特殊字符如空格&nbsp等等。

CSS语法由选择器、属性和值构成。选择器指定需要定义样式的元素，属性是具体要定义的样式，每个属性有一个值；CSS盒模型由content（内容）、padding（内边距）、border（边框）和margin（外边距）组成；盒模型布局主要有display（显示类型）、float（是否浮动）、flex（伸缩盒）等；CSS定位主要由position（定位类型）、上下左右、z-index（堆叠顺序）组成。此外就是各种纷繁复杂的元素样式，此处不再赘述。

JavaScript的标准是ECMAScript，最新为ES6。ECMAScript，描述了该语言的语法和基本对象。文档对象模型（DOM），描述处理网页内容的方法和接口。 浏览器对象模型（BOM），描述与浏览器进行交互的方法和接口。主要功能有嵌入动态文本于HTML页面、对浏览器事件做出响应、读写HTML元素、 检测访客的浏览器信息、控制cookies等。JavaScript是一种解释性脚本语言（代码不进行预编译），主要用来向HTML页面添加交互行为并可以直接嵌入HTML页面。

心得体会：

从一开始只用Html搭建网站整体结构，到用CSS为其添加丰富多彩的样式，再到用js添加各种事件，由静到动，感觉自己的网站慢慢‘活’了过来，这令我无比自豪与喜悦。网站开发初期主要参照Bilibili的设计，在模仿中逐步学习成长，到后面逐步变得熟练，并能够自己设计。Bilbili的代码纷繁复杂，一个元素通常嵌套七八层div。我通过自己的理解，只用少量代码就能够基本复刻出它的效果，这也让我非常喜悦。但由于时间有限，还有很多复杂的设计想要实现但是没有来得及做，这不免有些许遗憾。日后的学习工作中可能以框架的使用为主，不会像这样一点一点搭建网站。但对于这些内容的学习无疑能够帮助我更好地使用、学习、理解框架。

前端技术发展：

2009AngularJS1.0版本以及Node诞生，其中最为核心的MVVM、模块化、自动化双向数据绑定、语义化标签、依赖注入等等，为前端工程化解决了提供了一个可行的解决方案。2011年react诞生，2014年Vue.js诞生。自此，前端进入3.0工程化时代。如今Vue、React、Angular 三大框架并驾齐驱。基于jquery频繁操作DOM消耗性能等缺陷，Vue跟React均采用了虚拟dom这一思想，从而有效的解决频繁操作dom会损耗性能这个问题。另一个方面，随着node的出现，让前端工程化成为了现实。我们可以通过webpack这样的工具，对特定的语法进行转义，对高阶的语法编译成浏览器支持的es5语法，对页面进行组件开发，对代码进行分包，压缩等等一系列的处理。

如今，多元化的前端框架使这门行业兴起了其独特的工业时代，诸多科技百花争放、百家争鸣。