

项目四

CSS3 构建 Web 项目首页

教学导航

- 掌握 favicon 图标的制作，能够为 Web 项目制作 favicon 图标。
- 掌握 CSS3 新增选择器，能够实现网页元素精确控制。
- 掌握 Web 字体设置，能够在 CSS 文件中使用 Web 字体库中的字体。
- 掌握 CSS3 新增单位，能够提升网页中字体的可访问性和兼容性。
- 了解 Bootstrap 框架技术。

CSS3 是 CSS 技术的升级版本，主要包括盒子模型、列表模块、超链接方式、语言模块、背景和边框、文字特效、多栏布局等模块。CSS3 更加灵活，也更加复杂。本项目主要介绍如何运用 CSS3 构建 Web 项目中的图标、字体等。

项目描述

电商网站是企业为客户提供的访问内部资源的统一平台。这类网站必须具有直观的布局和吸引人的设计，以展示商品和引导用户进行浏览和购物。

本项目为设计时尚驿站（fashionHub）项目首页，页面显示效果如图 4-1 所示。



图 4-1 时尚驿站项目首页

1. 项目需求

(1) 搜索引擎优化 (search engine optimization, SEO): 利用搜索引擎规则提高网站搜索排名, 确定关键词、竞品调研、架构调整、布局优化, 确定网站的整体设计风格, 包括色彩、排版、制作 favicon 图标等, 以展现品牌形象和吸引用户。

(2) 设计顶部导航栏和搜索栏: 在页面顶部创建导航栏, 包含主要的网站页面链接, 并在导航栏旁边添加搜索栏, 方便用户快速搜索感兴趣的商品。

(3) 展示商品分类: 在首页上显示商品分类, 让用户可以按类别浏览和筛选商品, 提供方便的导航方式。

(4) 推荐商品和热门商品: 在首页上展示推荐的商品、热门商品或特价商品, 以吸引用户点击和购买。

(5) 添加底部导航和页面链接: 在页面底部创建导航栏和相应的页面链接, 方便用户浏览和访问网站的其他页面。

(6) 优化页面加载速度和响应式设计: 确保首页具有良好的加载速度, 对图片进行优化, 并确保网站在不同设备上有最佳的显示和交互效果。

2. 项目实施技术分析

(1) 使用 HTML5 新增标签完成本项目的网页结构设计。

(2) 使用 CSS3 新增属性完成本项目的页面样式。

(3) 建议使用 PS 准备本项目页面图标文件。

(4) 采取结构与样式相分离的模块化开发思路。

(5) 利用 chrome 浏览器实现项目效果浏览测试。

3. 项目文件夹结构设计

清晰的目录结构能够提高项目的开发效率。项目的目录结构首先要易于阅读和维护, 比如简洁明了的目录名称; 其次还要易于扩展和模块化, 能够顺利支撑项目的后期扩展; 再次是保持规范和统一, 比如文件名称的单复数等。本项目 fashionHub 的项目文件夹结构及其说明见表 4-1。

表 4-1 fashionHub 项目文件夹结构

名称	说明
fashionHub	项目文件夹名称 (一级目录)
imgs	页面固定图片文件夹 (二级目录)
style	页面样式文件夹 (二级目录)
fonts	字体图标文件夹 (二级目录)
uploads	页面经常更新图片文件夹 (二级目录)
index.html	首页文件

其中，style 文件夹结构及说明见表 4-2。

表 4-2 style 文件夹

名 称	说 明
base. css	项目基础样式
common. css	项目通用样式
index. css	首页特定样式

任务一 制作项目 favicon 图标文件

任务说明

favicon 是 favorites icon（最喜欢的图标）的缩写，是网站在浏览器标签页、书签栏和收藏夹等地方显示的小图标。它是网站的标识符，主要用来增加品牌辨识度和提升用户体验。

本次任务是为时尚驿站项目首页设计并制作网站 favicon 图标。

知识导入

favicon 也称为 website icon（网页图标）、page icon（页面图标）或者 url icon（网址图标）。favicon 图标是显示在地址栏上网站的视觉标志，让网页更加突出。如果网站没有提供相应的 favicon 文件，Web 服务器就会调用一个 404 错误文件。为了避免这个问题，需要为网站提供一个 favicon 图标。

favicon. ico 文件不是标准的位图文件，因此需要借助专门的编辑器生成。通常，制作一个 favicon 图标主要包括以下几个步骤。

第一步：创建图标图像。确定简洁明了、易于识别的网站的标志、颜色方案等设计元素。使用图形编辑软件（Adobe Photoshop）设计符合项目风格的正方形图片，或者使用现有的照片、图片，通过 Favicon Generator 在线编辑器生成图标图像。通常，favicon 的建议尺寸是 16×16 像素或 32×32 像素，并且使用透明背景以适应不同的浏览器主题。

第二步：命名图标文件。将第一步中创建的图标图像保存为 .ico 格式，并命名为 favicon. ico。

第三步：上传图标文件。将第二步中保存的 favicon. ico 文件上传到项目文件夹的根目录下，并确保此文件是唯一的。

第四步：引入图标文件，编辑项目文件夹中的 index. html 文档，在 <head></head> 标签内，添加如下代码引入 favicon 图标文件。

```
<link rel = " shortcut icon " href = "../favicon. ico " type = " image/x - icon ">
```

第五步：浏览图标文件。保存第四步中修改后的 HTML 文件，打开浏览器，刷新网站页面可以查看 favicon 图标的显示效果。

任务实现

1. 使用现有图片制作与项目相关的 logo。进入素材网站（<https://pixabay.com/>）选择并下载适合项目宣传的图片，保存至项目文件夹下的 imgs 文件夹内。

2. 打开比特虫（<https://www.bitbug.net>）在线生成器，单击“选择文件”按钮选择第一步中保存到 imgs 文件夹下的图像文件，点击下拉列表选择目标尺寸，输入网站提供的附加码，单击“生成 ico 图标”按钮，即可生成 favicon 图标文件，如图 4-2 所示。



图 4-2 favicon 图标的制作

3. 将生成的 favicon.ico 文件复制并粘贴到 fashionHub 项目文件夹根目录下，如图 4-3 所示。



图 4-3 fashionHub 项目文件夹

4. 将 favicon.ico 图标文件引入 fashionHub 项目文件夹的 index.html 文件中, 结合项目开发初期完成的关键词调研完成 SEO 优化。代码如下:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <!-- 网站 TDK 的 SEO 优化:
  TDK 即网站的标题(title)、描述(description)和关键词(keyword)。
  TDK 是网站很重要的元素,是爬虫爬取网站后最先看到的。
  1. title:用于设置搜索引擎识别网站归属的一个非常重要的入口标签。
      声明方式:网站名-不超 30 个汉字的网站主题-->
  <title>时尚驿站(fashionHub)-电商网站、时尚单品、正品低价、品质保障</title>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <!-- 2. description:网站业务和主题的概括。
      常采用“我们是……”“我们提供……”“电话:0312……”等语句-->
  <meta name="description" content="时尚驿站(fashionHub)-我们是专
业的电商网站,主要展示时尚活动、销售时尚单品,为用户提供愉快的购物体验。">
  <!-- 3. keywords:为搜索引擎提供网站搜索关键词。
      通常为 6~8 个关键词,关键词之间采用逗号分隔符 -->
  <meta name="keywords" content="时尚单品,时尚展示,名表,名包">
  <!-- 引入 favicon 图标图像 -->
  <link rel="shortcut icon" href="./favicon.ico" type="image/x-
icon">
</head>
<body>
</body>
</html>

```

5. 在浏览器查看 fashionHub 项目的 favicon 图标文件, 如图 4-4 所示。



图 4-4 带有 favicon 图标文件的网页

任务二 使用 CSS3 选择器实现页面元素精准布局

任务说明

首页顶部导航栏和搜索区在电商网站中通常有着非常关键的作用。它们需要具备明确、直观和易用的特点，以帮助用户快速导航和搜索所需的商品信息。

本次任务是为时尚驿站项目首页设计页面顶部导航栏和搜索区。将网站中主要的链接放置在顶部导航栏中；通过视觉上的高亮状态显示当前所在导航，帮助用户了解所在位置；使用下拉菜单显示和组织导航子页面链接；设计搜索栏。

知识导入

1. box-sizing 属性

盒子模型 (box model) 是 CSS 中用于页面元素布局的思维模型。在 HTML 中，每个元素都被渲染成一个矩形盒子，因此，Web 页面就是由一个一个矩形盒子构成的。

盒子模型具有以下四大属性。

(1) 内容区域 (content area)：实际显示内容的区域，是否对应 CSS 中的 width 属性取决于当前浏览器使用哪种盒子模型，例如，文本、图像或其他任何可见内容。

(2) 内边距 (padding)：内容区域与边框之间的空白区域，对应 CSS 中的 padding 属性。通过设置内边距，可以在内容区域周围创建额外的空间。

(3) 边框 (border)：围绕内容和内边距的线条或样式，对应 CSS 中的 border 属性。通过设置样式、宽度和颜色，可以创建不同的边框效果。

(4) 外边距 (margin)：盒子与其他盒子之间的空白区域，对应 CSS 中的 margin 属性，用于控制元素位置和元素之间的间距。

标准的 CSS 盒子模型如图 4-5 所示。

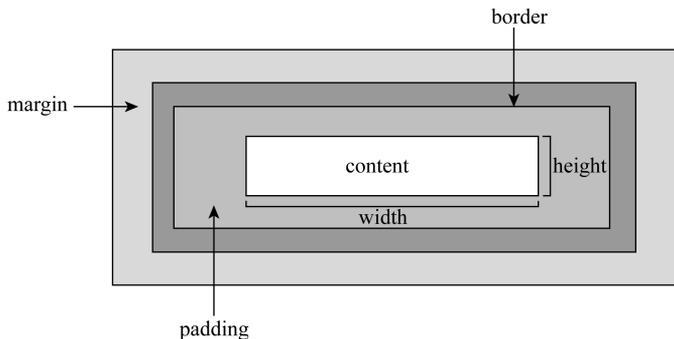


图 4-5 CSS 盒子模型

由于标准盒子模型和怪异盒子模型的存在，在不同盒子模型下的浏览器中，同一段 CSS 代码渲染的元素中 width 和 height 属性的含义却不相同。CSS3 为了解决不同浏览器中盒子模型引起的兼容问题，新增了 box-sizing 属性。

语法格式如下：

```
box-sizing:content-box | border-box | inherit;
```

其中，box-sizing 的属性值及其含义见表 4-3。

表 4-3 box-sizing 属性

属性值	描述
content-box	默认值，这是由 CSS2.1 规定的盒子模型。width/height 只包括内容的高度和宽度，不包括 padding、border 和 margin，即在宽度和高度之外绘制元素的内边距和边框
border-box	CSS3 新增设置，width/height 包括内容的高度和宽度、padding 和 border，但不包括 margin，即元素指定的任何内边距和边框都将在已设定的宽度和高度内进行绘制
inherit	规定从父元素继承 box-sizing 属性值

案例 4-1：“box-sizing:content-box;”的运用。

项目四源代码/fashionHub/example01.html

```
div.boxSizing{
    /* 设置 box-sizing 为默认值 */
    box-sizing:content-box;
    /* div 的总宽度:width+2* padding+2* border=200+10+4=214px */
    width:200px;
    height:200px;
    padding:5px;
    /* 当 border 为 1px 时,会有一个分辨率的计算偏差 */
    border:2px solid red;
    margin:3px;
}
```

在默认情况下，CSS 中元素的盒子模型采用的是“box-sizing:content-box”，即元素的宽度（或高度）值仅包括内容区域的大小，不包括边框、内边距和外边距。因此，上述代码中最终元素计算的结果为：div 元素总宽度 = width + 2 * padding + 2 * border = 214px（图 4-6）。

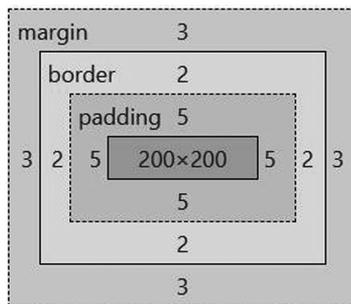


图 4-6 “box-sizing:content-box” 盒子模型

案例 4-2: “box-sizing:border-box;” 的运用。

项目四源代码/fashionHub/example02.html

```
div.boxSizing{
    /* 设置 box-sizing 值 */
    box-sizing:border-box;
    /* div 的总宽度:width=200px */
    width:200px;
    height:200px;
    padding:5px;
    /* 当 border 为 1px 时,会有一个分辨率的计算偏差 */
    border:2px solid red;
    margin:3px;
}
```

当设置 “box-sizing:border-box;” 时, 该元素的宽度 (或高度) 值将包括内容区域、边框和内边距 3 部分。因此, 上述代码中最终元素计算的结果为: div 元素总宽度 = width = 200px (图 4-7)。

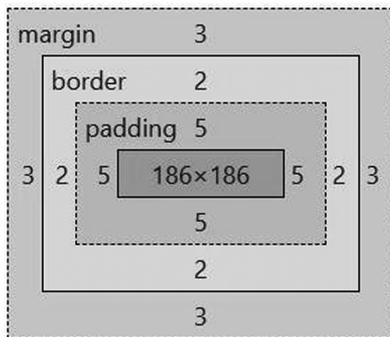


图 4-7 “box-sizing:border-box” 盒子模型

使用 border-box 盒子模型可以避免由于边框和内边距的添加导致元素尺寸不稳定变化的情况, 确保布局的准确性。在响应式布局或使用百分比单位进行尺寸计算时, border-box 盒子模型尤为重要。

但是, box-sizing 属性无论采取哪种设置, 元素总宽度 (或高度) 始终没有计算 margin 值。

思考: 如果项目中某元素必须增加外边距, 又该如何来精准控制该元素的布局呢?

2. calc() 函数

calc() 函数是 CSS3 的一个新增函数, 允许动态计算长度值, 经常与 margin 属性结合使用以实现动态调整外边距的效果, 这对于创建需要精确测量的复杂布局非常有用。

语法格式如下：

```
calc (expression) ;
```

其中，expression 中可以包含数字、+、-、*、/等运算符以及长度单位。

注意：运算符前后需保留一个空格。

案例 4-3：calc() 函数的运用。

项目四源代码/fashionHub/example03.html

```
article{
    background-color:#f2f2f2;
    /* 设置 box-sizing */
    box-sizing:border-box;
    /* 使用 calc() 函数实现抗干扰,即始终从父元素 33.33%宽度中减去 20 像素外边距,创建一个响应式布局,适应不同的屏幕尺寸 */
    width:calc (33.33% - 2*10px);
    /* 设置外边距干扰 */
    margin:10px;
    padding:20px;
}
```

通过 margin 属性和 calc() 函数的结合，开发人员可以根据特定条件动态设置元素的外边距。这对于实现自适应布局和响应式设计非常有用，开发人员能够根据不同的屏幕尺寸或布局要求对元素的外边距进行精确调整。

3. repeat() 函数

repeat() 函数也是 CSS3 的新增函数之一，是用于定义网格布局（grid layout）中重复模式的一个重要函数。repeat() 函数中包含重复次数和重复模式两个重要参数。该函数目前仅支持在网格容器的模板列（grid-template-columns）或模板行（grid-template-rows）中快速重复定义相同参数的操作。

语法格式如下：

```
repeat([auto-fill | auto-fit | <integer>] , [<line-names> ? [<fixed-size> | <track-size>]] + <line-names>?)
```

其中：

- auto-fill：表示根据容器的可用空间自动填充网格单元数量，让一行（或者一列）中尽可能容纳更多的单元格。
- auto-fit：与 auto-fill 类似，区别是会自动调整单元的大小以填充空间。
- <integer>：表示重复的次数。
- <line-names>：可以选择的任意名称。
- <fixed-size> | <track-size>：可以是长度值、百分比等，例如 fr、auto、max-content、min-content。

案例 4-4: repeat() 函数的运用。

项目四源代码/fashionHub/example04.html

```
.container {
  /* 设置 grid 布局 */
  display:grid;
  width:500px;
  background-color:#87f7a9;
  grid-gap:5px;
  /* repeat (重复次数,占用空间份数) */
  grid-template-columns:repeat(3,1fr);
  border:1px dashed rgb(235,238,239);
}
```

显示效果如图 4-8 所示。



图 4-8 repeat() 函数布局效果

4. 新增选择器

CSS3 在基本选择器、层级选择器的基础上又引入了新的选择器。CSS3 新增选择器拥有更精确的选择能力，减少了 HTML 结构的复杂性，为开发人员带来了许多便利。下面重点介绍属性选择器、子元素选择器、伪类选择器、伪元素选择器。

(1) 属性选择器

属性选择器 (attribute selector) 是根据网页中标签的属性名及属性值来实现精确选择的一种新增标签。该选择器由标签、中括号 “[] ” 构成，中括号内部是属性或者属性表达式，其结构如图 4-9 所示。

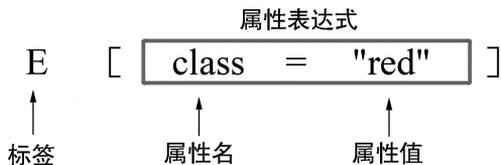


图 4-9 属性选择器结构

CSS3 中常用属性选择器及其描述见表 4-4。

表 4-4 常用属性选择器

属性选择器	描述	示例
E[attr]	选择具有指定属性名的元素	img[title]
E[attr = value]	选择与指定属性值完全匹配的元素	p[id = "one"]

续表

属性选择器	描述	示例
E[attr~ = value]	选择包含某指定属性值的元素	p[class ~ = "one"]
E[attr^ = value]	选择属性值以指定值开头的元素	a[href^ = "mailto;"]
E[attr\$ = value]	选择属性值以指定值结尾的元素	img[src \$ = "jpeg"]
E[attr * = value]	选择属性值中包含指定值的元素	a[title * = "site"]

案例 4-5：属性选择器的运用。

项目四源代码/fashionHub/example05.html

```
/* 用属性选择器设置属性 title */
.box[title="center"] { background-color:rgb(253,240,56); }
```

显示效果如图 4-10 所示。

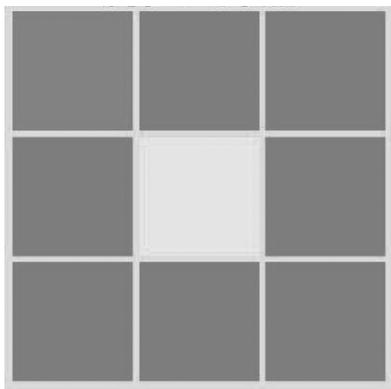


图 4-10 属性选择器布局效果

(2) 子元素选择器

子元素选择器 (child selector) 又称为结构化伪类选择器 (structural pseudo-class selector), 是 CSS 中一组用于选择元素在其结构排列中特定位置的选择器, 即将父元素中所有子元素基于其在文档结构中的位置、顺序和关系统一进行排序, 并按照该排序来选择目标元素。子元素选择器主要用于简化文档内标签 class 属性的定义。该选择器由标签、冒号“:”构成, 冒号后是伪类名表达式, 其结构如图 4-11 所示。

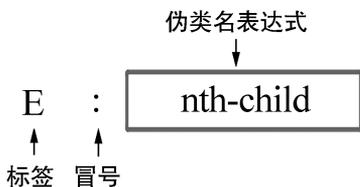


图 4-11 子元素选择器结构

CSS3 中常用子元素选择器及其描述见表 4-5。

表 4-5 常用子元素选择器

子元素选择器	描述
E:first-child	选择父元素下第一个子元素且该子元素为 E 的元素
E:last-child	选择父元素下最后一个子元素且该子元素为 E 的元素
E:nth-child(n)	选择父元素中的第 n 个子元素，元素类型没有限制
E:nth-of-type(n)	选择父元素中的第 n 个指定类型的子元素，元素类型没有限制

案例 4-6：子元素选择器的运用。

项目四源代码/fashionHub/example05.html（部分代码）

```
<style>
  /* 子元素选择器 */
  div:nth-child(2){
  /* 添加背景色 */
  background-color:rgb(113,30,238);
  /* 添加边框 */
  border:5px solid red;
  }
</style>
<body>
  <h1>属性选择器</h1>
  <div class="wrapper">
    <div class="box" id="one"></div>
    <div class="box"></div>
    <div class="box"></div>
    <div class="box"></div>
    <div class="box" title="center"></div>
    <div class="box"></div>
    <div class="box"></div>
    <div class="box"></div>
    <div class="box"></div>
  </div>
</body>
```

显示效果如图 4-12 所示。

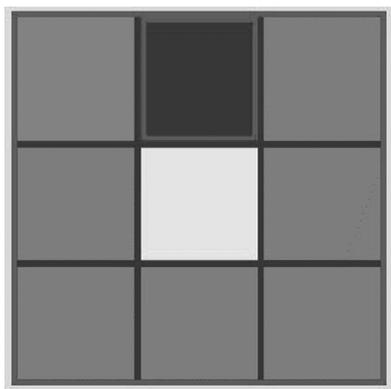


图 4-12 子元素选择器布局效果

思考：使用子元素选择器添加背景和边框后，为为什么样式会作用到 `div.wrapper` 和 `div.box` 两个元素上呢？

在上述代码中，通过子选择器定位文档中作为父元素第二个子元素且标签名为 `div` 的元素时，出现了两个符合条件的 `div`。第一个符合条件的 `div` 来自 `body` 作为父元素，`h1` 作为 `body` 的第一个子元素，`div.wrapper` 作为 `body` 的第二个子元素；第二个符合条件的 `div` 来自 `div.wrapper` 作为父元素，`div#one` 作为 `div.wrapper` 的第一个子元素，排在后续的 `div` 作为 `div.wrapper` 的第二个子元素。因此，改变颜色和添加背景的风格既运用到了 `div.wrapper` 元素上，又运用到了第二个小的 `div` 元素上。这就是出现上述样式效果的原因。

由此可见，子元素选择器是不区分元素类型的，在文档内如果不同类型元素混合编写在一起，就会出现定位不准确的问题。如何精准控制同一类元素呢？如下调整上述代码中的选择器。

```
div:nth-of-type(2){
    /* 设置背景色 */
    background-color:rgb(113,30,238);
    /* 设置边框 */
    border:5px solid rgb(239,94,181);
}
```

思考：`:nth-child()` 与 `:nth-of-type()` 分别是按照什么规则来排序的呢？

`:nth-child()` 在排序时是将父元素中全部元素（不区分 `type`）混合在一起排序，例如 `h1` 和 `div`。而 `:nth-of-type()` 则是过滤出全部父元素后再排序，因此，`div.wrapper` 就不符合本次筛选条件了，所以使用 `:nth-of-type()` 选择器实现了精准定位。

(3) 伪类选择器

伪类选择器（`pseudo-class selector`）是根据元素的状态选择特定的元素，从而实现针对状态的样式控制的一组选择器。它们增加了 CSS 的选择功能，通过使用伪类选择器，可

以实现更多样的效果和交互行为，提高用户体验和页面表现力。伪类选择器的结构同于元素选择器。

CSS3 中常用伪类选择器及其描述见表 4-6。

表 4-6 常用伪类选择器

伪类选择器	描述
E:hover	选择处于鼠标悬停状态的 E 元素
E:active	选择处于激活（如点击时）状态的 E 元素
E:checked	选择处于被选中状态的表单元素

案例 4-7：伪类选择器的运用。

项目四源代码/fashionHub/example05.html（部分）

```
/* 伪类选择器 */
div.box:hover{
    /* 当存在鼠标悬停状态时给当前元素加亮绿色 */
    background-color:lightgreen;
}
```

(4) 伪元素选择器

伪元素选择器（pseudo-element selector）是 CSS 中用于选择元素的特定部分或生成内容的选择器，主要用于向元素的特定位置添加样式或创建内容。该选择器以标签、双冒号“::”开头，双冒号后是伪元素名，其结构如图 4-13 所示。

E :: before
 ↑ ↑ ↑
 标记 双冒号 伪元素名

图 4-13 伪元素选择器结构

CSS3 中常用伪元素选择器及其描述见表 4-7。

表 4-7 常用伪元素选择器

伪元素选择器	描述
E::before	在 E 元素之前插入内容
E::after	在 E 元素之后插入内容

案例 4-8：伪元素的运用。

项目四源代码/fashionHub/example05.html

```
/* 结合 body 中的 CSS 计数器 */
.box::before{
```

```

/* content 必填属性,显示的内容。
自行设置显示格式,如:点号+空格 counter(mycounter) ". " * /
content:counter(mycounter);
/* 设置计数器计算方式 */
counter-increment:mycounter;
}

```

显示效果如图 4-14 所示。

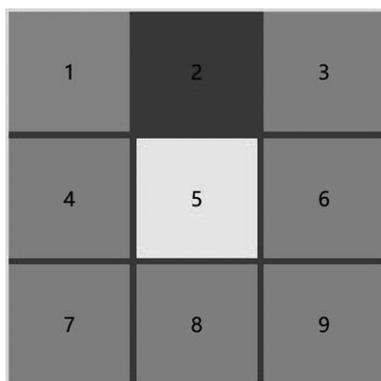


图 4-14 伪元素选择器布局效果

5. 选择器权重

在 CSS 中,样式列表所在位置的不同,对元素的影响会不同,其优先级为:内联样式 (style 属性) ><style>标签>和<link>标签。进入具体样式列表内部后,基于减少遍历次数的原则,浏览器会针对每一组选择器从右向左进行匹配解析,当多个规则同时适用于一个元素时,选择器与选择器之间根据权重确定样式规则应用的优先级。

选择器权重由 6 部分组成,优先级顺序见表 4-8。

表 4-8 选择器优先级

选择器权重 (specificity)	描述
无穷大	!import, 覆盖当前页面所有相同元素样式
1000	在元素标签内直接添加 style 属性, 即内联样式
100	ID 选择器 (#div1)
10	类选择器、属性选择器或伪类选择器 (.red、[type="text"]、:hover)
1	元素选择器或伪元素选择器 (div、::before)
0	通用选择器 (*)

选择器权重是十进制的，但不能进位，也可以看作是 1, 1, 1 形式。例如，选择器 `div. foo` 的权重值是 011，因为它含有一个元素选择器（1）和一个类选择器（10）。

在确定最终优先级时，CSS 会根据每个规则的权重进行比较。如果有多个规则具有相同的权重，则后面的规则将覆盖前面的规则；如果两个或多个规则具有不同的权重，具有较高权重的规则将优先应用。

案例 4-9：权重的应用。

项目四源代码/fashionHub/example06. html

```
#container { /* 仅有一个 id 选择器，权重为 100 */
  color:red;
}
body div#container { /* 两个元素选择器，一个 id 选择器，权重为 102 */
  color:blue;
}
div.container { /* 一个元素选择器，一个类选择器，权重为 011 */
  color:green;
}
.container { /* 仅有一个类选择器，权重为 010 */
  color:purple;
}
```

在上述代码中，`body div#container` 样式规则的权重最高，为 102；其次是 `#container` 样式规则，权重为 100；排在第三位的是 `div.container` 样式规则，权重为 011；`.container` 样式规则具有最低的权重，为 010；因此，文本的最终颜色为蓝色。

理解选择器权重对于理解 CSS 样式继承和优先级非常重要，可以帮助开发人员更好地控制样式的应用和覆盖。

任务实现

1. 编辑 `index.html`，引入 `style` 文件夹下 CSS 样式文件。代码如下：

```
<!-- 引入基础 CSS 样式 -->
<link rel="stylesheet" href="./style/base.css">
<!-- 引入通用 CSS 样式 -->
<link rel="stylesheet" href="./style/common.css">
```

2. 编辑 `index.html` 文件，设计网页结构。（详见 `fashionHub/index.html` 文件。）

3. 编辑基础样式 `base.css` 文件，实现页面样式初始化。代码如下：

```
/* 项目页面样式初始化:统一边距、文字、边框等 */
/* 通配符选择器清除边距 */
* {
```

```
margin:0;
padding:0;
/* 页面全部采取css3盒子模型 */
box-sizing:border-box;
}
/* 统一去掉li的小圆点 */
li {
    list-style:none
}
/* 统一去掉img边框 */
img {
    /* 防止图片边框问题 */
    border:0;
    /* 将图片或表单元素(行内块)和文字垂直居中 */
    vertical-align:middle;
}
/* 鼠标经过时显示小手图标 */
button {
    cursor:pointer
}
/* 统一取消链接下划线 */
a {
    color:#666;
    text-decoration:none
}
/* 伪类选择器,统一设置悬停状态文字颜色 */
a:hover {
    color:#c81623
}
/* 统一去掉默认框线 */
input {
    border:0;
    outline:none;
}
body {
    /* 为整个项目网页设置统一字体,优化浏览器兼容效果。
    例如:
```

```
    小米:Helvetica Neue, Helvetica, Arial, Microsoft Yahei, Hiragino
Sans GB, Heiti SC, WenQuanYi Micro Hei, sans-serif;
    京东:Microsoft YaHei, Heiti SC, tahoma, arial, Hiragino Sans GB, "
\5B8B\4F53", sans-serif;
    天猫:font-family:"PingFang SC", miui, system-ui, -apple-system,
BlinkMacSystemFont, Helvetica Neue, Helvetica, sans-serif;
    Github:font-family:-apple-system, BlinkMacSystemFont, Segoe UI,
Helvetica, Arial, sans-serif, Apple Color Emoji, Segoe UI Emoji, Segoe UI
Symbol;
    * /
    font-family:Microsoft YaHei, Heiti SC, "\5B8B\4F53";
    /* 控制字体渲染时的平滑效果 */
    -webkit-font-smoothing:antialiased;
    -moz-osx-font-smoothing:grayscale;
    /* 统一页面背景色 */
    background-color:#fff;
    /* 统一页面字体大小,为响应式做准备 */
    font:14px Microsoft YaHei, Heiti SC, "\5B8B\4F53";
    /* 统一文字颜色 */
    color:#666
}
```

4. 编辑 common.css 文件, 实现通用样式。代码如下:

```
/* 顶部导航栏 */
.topnav {
    height:31px;
    line-height:31px;
    background-color:#f1f1f1;
}
/* 主体结构 */
/* 类选择器的使用 */
.container {
    /* PC 端宽度 */
    width:1200px;
    /* 居中设计 */
    margin:0 auto;
}
```

```
.left-content {
    float:left;
}.right-content {
    float:right;
}
}
}.style-red {
    color:#c81623;
}
}.topnav ul li {
    float:left;
}
/* 子元素选择器 */
}.topnav .right-content ul li:nth-child(even) {
    width:1px;
    height:12px;
    background-color:#666;
    margin:9px 15px 0;
}
/* header 头部搜索栏 */
}.header {
    position:relative;
    height:105px;
}
}.search {
    position:absolute;
    left:346px;
    top:25px;
    width:538px;
    height:36px;
    border:2px solid #b1191a;
}
}.search input {
    float:left;
    width:454px;
    height:32px;
    padding-left:10px;
```

```
}  
.search button {  
    float:left;  
    width:80px;  
    height:32px;  
    background-color:#b1191a;  
    font-size:16px;  
    color:#fff;  
}  
.hotwords {  
    position:absolute;  
    top:66px;  
    left:332px;  
}  
.hotwords a {  
    margin:0 10px;  
}  
.shopcar {  
    position:absolute;  
    right:60px;  
    top:25px;  
    width:140px;  
    height:35px;  
    line-height:35px;  
    text-align:center;  
    border:1px solid #dfdfdf;  
    background-color:#f7f7f7;  
}  
.count {  
    position:absolute;  
    top:-5px;  
    left:105px;  
    height:14px;  
    width:25px;  
    line-height:14px;  
    color:#fff;  
    background-color:#e60012;
```

```
padding:0 5px;
border-radius:7px 7px 7px 0px;
}
/* 主体导航 */
.nav {
    height:47px;
    border-bottom:2px solid #b1191a;
}
.nav .nav-dropdown {
    float:left;
    width:210px;
    height:45px;
    background-color:#b1191a;
}
.nav .nav-items {
    float:left;
}
.nav-dropdown .dt {
    width:100%;
    height:100%;
    color:#fff;
    text-align:center;
    line-height:45px;
    font-size:16px;
}
.nav-dropdown .dd {
    display:none;
}
.nav-dropdown: hover .dd{
    width:210px;
    height:465px;
    background-color:#c81623;
    margin-top:2px;
    display:block;
}
.nav-dropdown .dd ul li {
    position:relative;
```

```
    height:31px;
    line-height:31px;
    margin-left:2px;
    padding-left:10px;
}
.nav-dropdown.dd ul li:hover {
    background-color:#fff;
}
.nav-dropdown.dd ul li a {
    font-size:14px;
    color:#fff;
}
.nav-dropdown.dd ul li:hover a {
    color:#c81623;
}
.nav-items ul li {
    float:left;
}
.nav-items ul li a {
    display:block;
    height:45px;
    line-height:45px;
    font-size:16px;
    padding:0 25px;
}
/* 底部模块制作 */
.footer {
    height:415px;
    background-color:#f5f5f5;
    padding-top:30px;
}
.footer-service {
    height:80px;
    border-bottom:1px solid #ccc;
}
.footer-service ul li {
    float:left;
```

```
        width:300px;
        height:50px;
        padding-left:35px;
    }
    .service h4 {
        font-size:14px;
    }
    .footer-help {
        height:185px;
        border-bottom:1px solid #ccc;
        padding-top:20px;
        padding-left:50px;
    }
    .footer-help dl {
        float:left;
        width:200px;
    }
    .footer-help dl:last-child {
        width:90px;
        text-align:center;
    }
    .footer-help dl dt {
        font-size:16px;
        margin-bottom:10px;
    }
    .footer-help img{
        width:60px;
    }
    .footer-copyright {
        text-align:center;
        padding-top:20px;
    }
    .links {
        margin-bottom:15px;
    }
    .links a {
        margin:0 3px;
```

```
}  
.copyright {  
    line-height:20px;  
}
```

任务三 制作 Web 字体提升用户体验

任务说明

目前，文字信息仍是网站中最主要的内容，直接影响用户对网站的初步印象和用户体验。使用适当风格的文字，不但可以吸引特定目标用户的注意力，还能够得到用户的情感认同。

本次任务是为时尚驿站项目首页设计 Web 字体，提升网站的用户体验。

知识导入

1. Web 字体格式

下面介绍几种常用的 Web 字体格式。

(1) TrueType 字体

TrueType 字体是 Windows 和 macOS 操作系统最常用的字体格式，文件扩展名为 .ttf。它是一种基于轮廓技术的字体，由指令描述字形，与分辨率无关，适合屏幕显示和打印输出。和矢量字体一样，它可以随意缩放、旋转而不必担心出现锯齿，能够保证字体在各种平台具有正确的外观以及屏幕显示与打印输出的一致性。

(2) OpenType 字体

OpenType 是 Microsoft 和 Adobe 公司联合开发一种字体格式，文件扩展名为 .otf。它也是一种基于轮廓技术的字体，是 TrueType 的超集。它具有更强的跨平台功能，生成的文件尺寸更加小巧，比 TrueType 更为强大。

(3) 嵌入式 OpenType 字体

嵌入式 OpenType 字体（Embedded OpenType，简称 EOT）是 Microsoft 公司设计的一种 IE 专用字体格式，这种字体文件扩展名通常为 .eot，是 OpenType 的压缩格式，它使用数字版权管理技术防止在未经许可的情况下被使用。

(4) Web 开放字体格式

Web 开放字体格式（Web Open Font Format，简称 WOFF）是一种专门为 Web 设计的字体，本质上是对 TrueType 和 OpenType 字体的压缩封装。

(5) SVG 字体

SVG 是由 W3C 制定的开放标准的矢量图形格式的字体。SVG 字体就是使用 SVG 技术来呈现的字体，目前只有 Safari 和 Android 浏览器支持。

2. 下载 Web 字体

下载免费的 Web 字体很快，也很方便，很多网站都提供免费字体下载，例如 Abstract Fonts (<https://www.abstractfonts.com/>)、Font Squirrel (<http://www.fontsquirrel.com/>) 等，如图 4-15 所示。

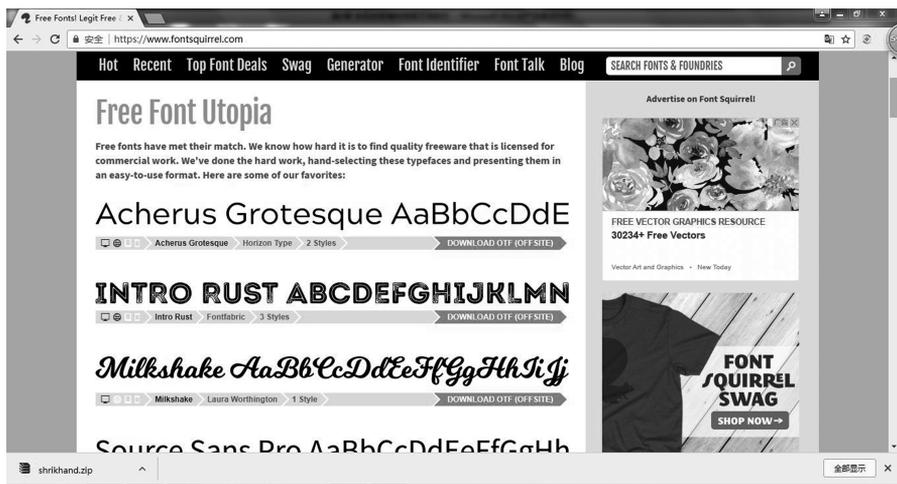


图 4-15 Web 字体下载

这里我们下载 Shrikhand 字体。点击“DOWNLOAD”超链接，开始下载。下载的文件是一个 ZIP 的压缩包。下载完成后，打开压缩包，就会看到一个 Web 字体文件，如图 4-16 所示。双击该 Web 字体文件就可以看到该字体的示范，如图 4-17 所示。



图 4-16 下载的字体压缩包



图 4-17 字体示范

3. 使用 Web 字体

在 CSS3 中，可以使用 @font-face 属性来调用服务器端字体。@font-face 属性的基本语法格式如下：

```
@font-face{
    font-family:"Web 字体名称";
    src:url("Web 字体文件") format("字体文件的格式");
}
```

其中：

- font-family：用来声明自定义的字体名称，该名字可以任意取，但最好是使用下载的字体默认名称。

- src：指定 Web 字体的字体文件所在的路径。

- format：声明字体文件的格式，可以省略文件格式的声明而单独使用 src 属性。

注意：TrueType、WOFF 字体对于 IE 浏览器来说，IE9 及以上版本才支持，IE8 只支持 EOT 格式的字体文件并且需要配合 format 验证。

案例 4-10：使用 Web 字体。

项目四源代码/fashionHub/example07.html

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>使用 Web 字体</title>
```

```

<style>
  @font-face {
    font-family:"Webfont-text";
  /* 在引入字体时的路径写法要非常注意:
    When using @font-face to declare multiple font types for cross browser compatibility, you can see 404's in old versions of IE due to a bug in the way that IE parses the font declarations. For example, this syntax:
    src:url('myfont-webfont.eot') format('embedded-opentype'),
    可改用下列写法:
    Will cause a 404 in IE 6,7,and 8. The fix is to add a question mark after the first font URL, so IE sees the rest of the property value as a query string. This is a correct example:
    src:url('myfont-webfont.eot? #iefix') format('embedded-opentype'), * /
    src:url("font/Shrikhand-Regular.otf") format("OpenType");
  }
  p{
    font-family:"Webfont-text";
  }
</style>
</head>
<body>
  <h1>HeadLines are very important</h1>
  <p>When you're a designer on the web, your portfolio site is the first and best chance you get to demonstrate your talent. Of course, we think it's even better if you also take the opportunity to show off your refined taste for typography. This week, we'll look at a few portfolio sites that accomplish both of these.</p>
</body>
</html>

```

在上述代码中，通过@font-face 规则定义了 Web 字体，<p>标签中的文本用了该 Web 字体。显示效果如图 4-18 所示。

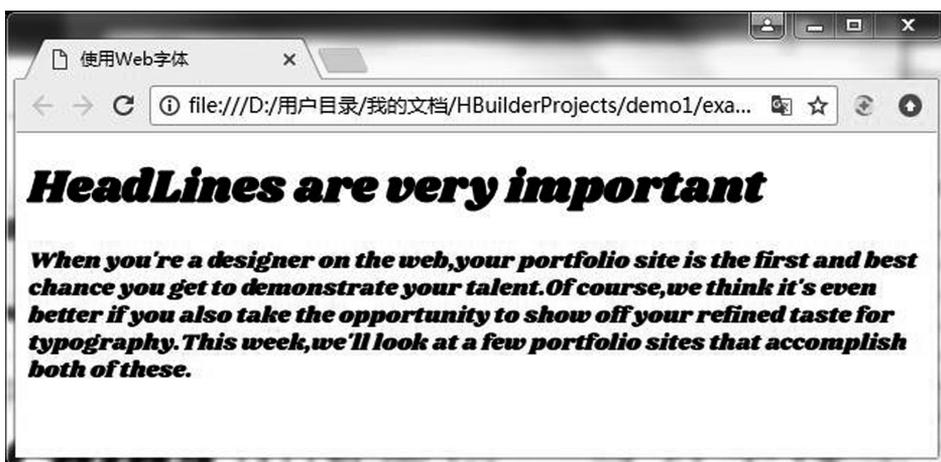


图 4-18 使用 Web 字体显示效果

4. 制作 Web 字体图标

除了创意字体外，为了提高网页的可视性、易用性和美观性，通常还会用到表示特定功能、导航菜单、社交媒体链接等的小型图像或符号。如果直接使用 image 图标往往会出现失真，所以，借助 icomoon 在线设计并制作 Web 字体图标在前端开发中也是非常重要的一项技能。

(1) 进入 icomoon 官网 (<https://icomoon.io>)，然后点击页面右上角的“Ico Moon App”按钮，进入 icomoon 字体图标库，如图 4-19 所示。

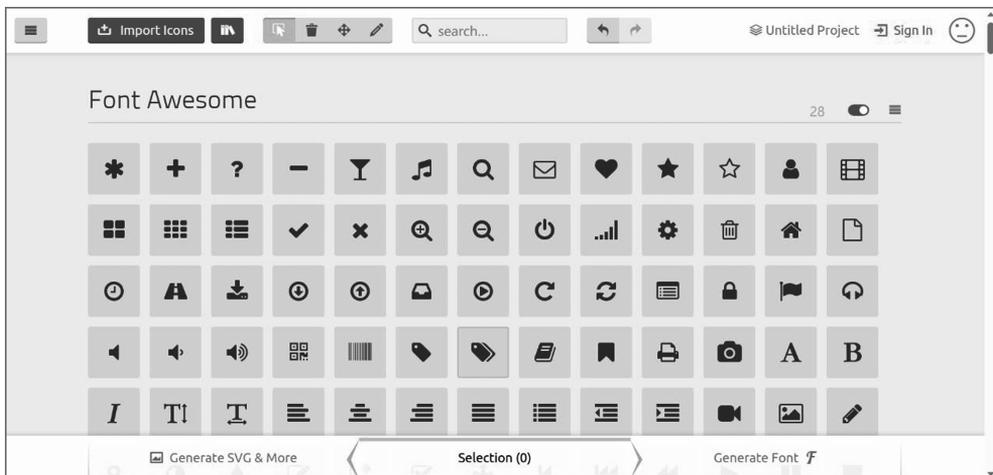


图 4-19 icomoon 字体图标库

(2) 新建自定义图标项目。单击左上角  按钮，在下拉列表中选择“New Empty Set”新建空项目（图 4-20），点击主页中出现的“Untitled set”最右侧的  图标，在下

拉列表中选择“Properties”（图 4-21），在属性设置弹窗（图 4-22）中，单击“Edit Metadata”按钮，进入属性详情页，更改“Name”输入框项目名，如“fashionHub”，单击右上角“关闭”按钮（图 4-23）。

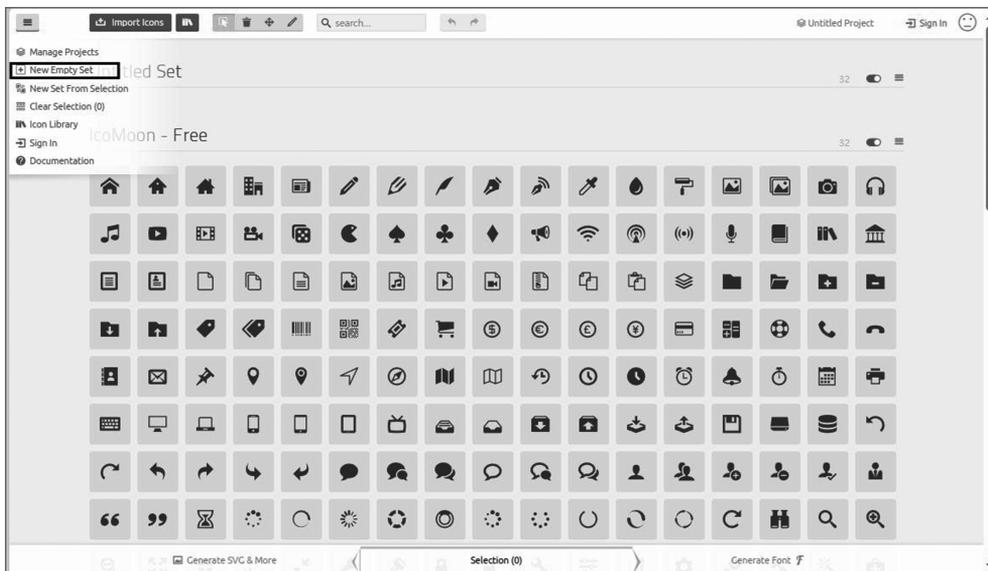


图 4-20 选择“New Empty Set”选项

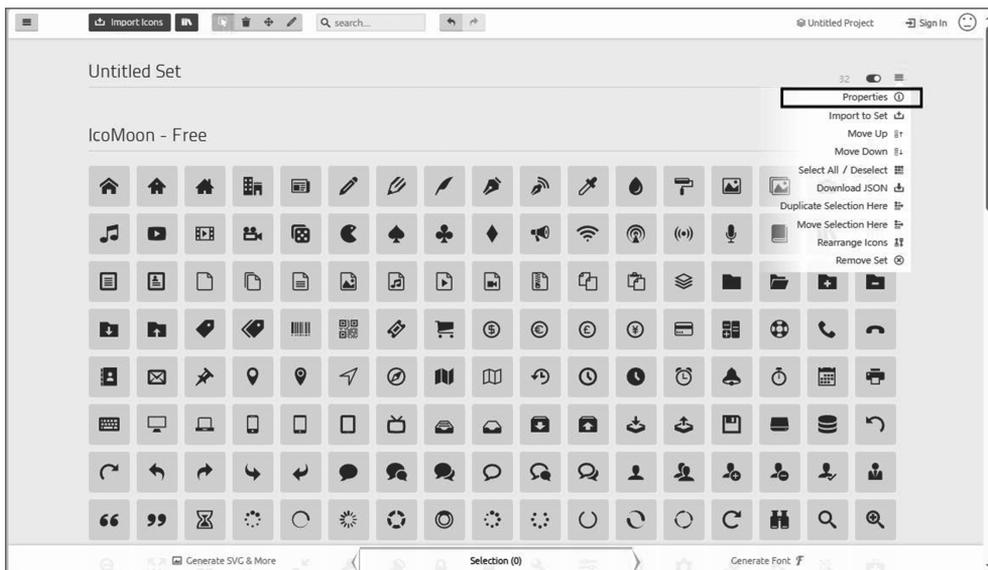


图 4-21 选择“Properties”选项

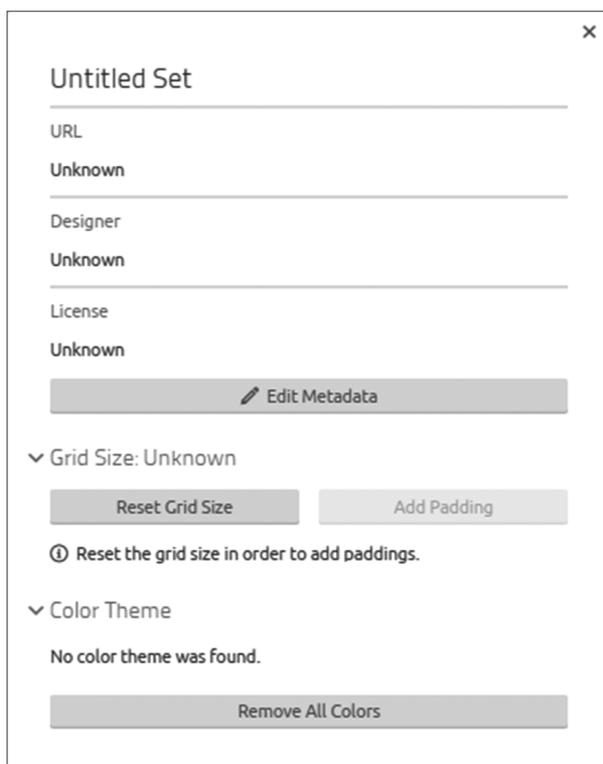


图 4-22 属性设置弹窗



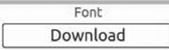
图 4-23 属性详情页

(3) 选取或制作所需字体图标并保存。

- 选取图标。icomoon 默认会提供图标，鼠标单击选中所需字体图标。如果默认提供

的字体图标中没有适合的图标，可以滚动到主页最下方单击“Add Icons From Library…”进入更多的字体图标库进行选择。

• 制作图标。单击页面左上角  导入外部 SVG 文件，可以是自己设计制作的 SVG 文件，也可以通过阿里巴巴矢量图库下载 SVG 文件。单击主页中项目“fashionHub”最右侧图标，选择“Duplicate Selection Here”，将选中的图标复制到项目中。

(4) 生成图标并下载。单击主页最下方 **Generate Font**  按钮进入字体图标页，将选中的图标生成字体，单击  下载并保存到本地 Web 项目文件夹，如 fashionHub/fonts/icomoon.zip。解压 icomoon.zip 后应包括以下文件。

- demo-files, demo.html: demo 示例文件。
- fonts: 字体文件。
- Read Me.txt: 说明文件。
- selection.json: iconmoon 项目文件，可用于导入。
- style.css: 项目中需要用到的样式文件。

任务实现

1. 编辑项目 fashionHub 文件夹下首页文件 index.html。代码如下：

```
<!-- 引入通用 CSS 样式 -->
<link rel="stylesheet" href="./style/common.css">
<!--或者直接引入字体图标文件夹 icomoon 自带样式文件 style.css -->
<link rel="stylesheet" href="./fonts/icomoon/style.css">
```

2. 编辑项目 fashionHub 文件夹下的 style/common.css。

复制 icomoon/style.css 中 @font-face 到 style/common.css（如果第一步使用的是直接引入 icomoon/style.css，则忽略本步骤）。代码如下：

```
/* 使用 Web 字体 */
@font-face {
    font-family:'icomoon';
    /* 考虑 IE8 及以下版本的兼容问题,url 的写法 */
    src:url('../fonts/icomoon/fonts/icomoon.eot? a7gvbu');
    src:url('../fonts/icomoon/fonts/icomoon.eot? a7gvbu #iefix')
format('embedded-opentype'),
    url('../fonts/icomoon/fonts/icomoon.ttf? a7gvbu') format
('truetype'),
```

```
        url ('../fonts/icomoon/fonts/icomoon.woff? a7gvbu ') format
('woff'),
        url ('../fonts/icomoon/fonts/icomoon.svg? a7gvbu #icomoon ')
format('svg');
    font-weight:normal;
    font-style:normal;
    font-display:block;
}
/* 主体结构 */
/* 伪元素选择器添加图标 */
.arrow-icon::after {
    content:"\f078";
    font-family:'icomoon';
    margin-left:6px;
}
/* 伪元素选择器 */
.shopcar::before {
    content:'\e900';
    font-family:'icomoon';
    margin-right:5px;
    color:#b1191a;
}
/* 主体导航 */
.nav-dropdown.dd ul li::after {
    position:absolute;
    top:1px;
    right:10px;
    color:#fff;
    font-family:'icomoon';
    content:'\e920';
    font-size:14px;
}
```

任务四 使用 CSS3 相对单位解决适配问题

任务说明

随着浏览设备的快速更新换代，PC 端 UI（界面）设计师，在做 UAT（user acceptance test，用户验收测试）时，也越来越多地遇到界面还原度问题。例如文字大小、图片大小、页面元素之间比例失衡，设计稿在更换屏幕尺寸以及设备像素差异等引起的整张页面样式调整。同样的页面内容，要在大小各异的屏幕上都能呈现出满意的效果，只能编写不同的网页，同时也要维护不同的版本。

本次任务为使用 CSS3 引入的相对单位优化时尚驿站项目首页样式，完成动态适配设计，确保页面在不同浏览设备上都能正确显示。

知识导入

CSS 中的单位是用来确定属性值大小的一种方式。CSS 支持很多单位，包括时间单位、长度单位等。其中长度单位的运用在定位与布局中是非常重要的环节。长度单位又分为绝对单位和相对单位。绝对单位是指在任何屏幕上都保持相同的长度，如像素（px）；而相对单位则是相对于其他元素进行计算，如百分比（%）。CSS3 在原有这些单位基础上又引入了几个相对单位，使得在不同设备上能更加灵活地实现适配。

1. em 单位

em 是 emphasize 的缩写，em 是一个相对长度单位，指相对于当前标签内文本 font-size 设置的大小。如果当前标签内文本尺寸没有人为设置，则是相对于其父元素 font-size 设置的大小；父元素也没有设置的，则是相对于浏览器默认字体大小。

案例 4-11：em 单位的运用。

项目四源代码/example08.html

```
body{
    font-size:20px; /* px 单位 */
}
#app{
    width:2em; /* 本身没有 font-size,因此继承 body */
    height:2em; /* 2x20=40px */
}
```

显示效果如图 4-24 所示。



图 4-24 继承父元素 font-size 的 em 单位

在上述代码中，div#app 标签内没有设置 font-size 属性，因此，使用 em 单位继承其父元素 body 中字体的大小。

思考：如果 div#app 标签内设置了 font-size 属性后，其宽度和高度分别是多少呢？

案例 4-12：默认 font-size 的运用。

项目四源代码/example09.html

```
body{
    font-size:62.5%;/* 62.5%x 16px=10px 浏览器默认 16px */
}
#app{
    width:2em; /* 本身没有 font-size,因此继承 body 2x10=20px */
    height:2em;
}
```

显示效果如图 4-25 所示。



图 4-25 父元素 font-size 使用百分比的 em 单位

在上述代码中，div#app 父标签内 font-size 属性为百分比，因此，需要借助浏览器默认文本大小首先计算出 body 中的文本大小。

思考：嵌套标签内使用 em 单位时如何计算长度？

2. rem 单位

rem 是 root em 的缩写，这个单位表示相对于 html 根元素的字体大小进行计算。在响应式设计中，rem 通常更受青睐，因为它可以更精确地控制元素的尺寸，而不受父元素字体大小的影响。但是，在某些情况下，如文本内部的字体大小设置，使用 em 可能更合适，因为它可以相对于父元素进行缩放。

需要注意的是，旧版的 IE（IE8 及以下版本）不支持 rem 单位，因此在使用时要考虑到浏览器的兼容性。

案例 4-13：rem 单位的运用。

项目四源代码/example10.html

```
html{
    font-size:16px;
}
body{
    font-size:30px;
}
#app{
    width:2rem; /* 2x16=32px */
    height:2rem;
}
```

显示效果如图 4-26 所示。



图 4-26 rem 单位的基本应用

在上述代码中，div#app 标签的宽度和高度使用 rem 单位后，则根据 html 根元素中 font-size 属性计算。

思考：改变 body 中 font-size 属性对 div#app 的宽度和高度有影响吗？如果 html 中没有设置 font-size，那么 div#app 的宽度和高度分别是多少呢？

3. vw 与 vh 单位

vw、vh 单位又称为视口单位，分别代表相对于浏览器视口宽度和高度的百分比，1vw = 视口 (viewport) 宽度 (width) / 100。1vh = 视口 (viewport) 高度 (height) / 100。在浏览器中，1vw 和 1vh 与百分比不同的是，vw 和 vh 始终是相对于可视窗口的宽度和高度，而百分比与其父元素的宽度和高度有关。

案例 4-14：vw 和 vh 单位的运用。

项目四源代码/example11.html

```
/* 检测 1vw */
.p1{
    width:1vw;
}
/* 检测 1vh */
.p2{
    width:1vh;
}
```

```
.bgimg{
    position:relative;
    background:url('./imgs/全屏背景图片.webp') center/cover;
    width:100%;
    /* width:100vw; */
    height:100vh;
    /* height:100%; */
}
```

显示效果如图 4-27~4-29 所示。

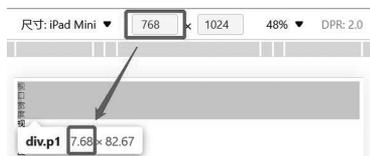


图 4-27 1vw 单位



图 4-28 1vh 单位



图 4-29 vw/vh 全屏背景

在大部分情况下，视口单位（vh 和 vw）和百分比单位都可以实现相似的功能。浏览器通常是根据浏览器窗口大小来计算视口大小，其中包括了滚动条所占用的空间。如果页面内容超出视口宽度，则视口宽度将大于 html 元素的宽度；同理，如果页面中某些内容需要在高度上占满整个屏幕，那么它必须保证不受其父标签的影响。

在上述代码中，背景图的宽度和高度分别采用了百分比与视口单位，思考一下，为什么宽度使用百分比，而高度使用视口单位 vh 呢？

CSS3 中引入的这些相对单位在响应式设计中非常有用，它们可以根据浏览器窗口的尺寸和字体大小进行自适应，确保元素在各种设备上得到良好的布局效果和可读性。

任务实现

1. 编辑 fashionHub/index.html 替换原有样式。代码如下：

```
<!-- 引入响应式 CSS 改写样式替换原有样式 common.css -->
<link rel="stylesheet" href="./style/rem-common.css">
```

2. 编辑 fashionHub/style/base.css，声明根元素字体大小。代码如下：

```
/* 在设计之初考虑好扩大或缩小比例,明确根元素字体大小 */
html{
    font-size:62.5%;
}
```

分析：浏览器默认字体为 16px，1rem=16px。显然，如果页面中其他设置全部采取整数倍，如 2rem、5rem，会导致页面元素太大，因此，1.2rem、1.5rem 就会频繁出现。但是，1.2rem=1.2×16=19.2px，这是不合理的。针对上述问题，开发人员常采取的解决方案是 16px×62.5%=10px，1.2rem=12px。

3. 编辑 fashionHub/style/rem-common.css，改写相对单位，实现根据根元素字体大小动态调整页面样式。代码如下：

```
/* 部分,详见 fashionHub 项目源代码 */
/* 顶部导航栏 */
.topnav {
    height:3.1rem;
    line-height:3.1rem;
    background-color:#f1f1f1;
}
/* 主体结构 */
/* 类选择器的使用 */
.container {
    /* PC 端宽度 */
    width:90rem;
    /* 居中设计 */
    margin:0 auto;
}
...
/* 子元素选择器 */
.topnav .right-content ul li:nth-child(even) {
    width:0.1rem;
    height:1.2rem;
    background-color:#666;
    margin:0.9rem 1.5rem 0;
}
/* 伪元素选择器添加图标 */
.arrow-icon::after {
    content:"\f078";
    font-family:'icomoon';
    margin-left:0.6rem;
}
```

```
/* header 头部搜索栏 */  
.header {  
    position:relative;  
    height:10.5rem;  
}  
...
```

任务五 知识拓展——认识 Bootstrap 框架

Bootstrap 是特推（Twitter）公司推出的基于 HTML、CSS 与 JavaScript 开发的 CSS 开发框架。它提供了一系列的 CSS 和 JavaScript 组件，用于开发响应式布局、移动设备优先的 Web 项目。Bootstrap 一经推出颇受欢迎，一直是 GitHub 上的热门开源项目。

1. Bootstrap 框架的特点与功能

Bootstrap 之所以受欢迎，是因为它占用资源少，样式、组件和插件多且易于掌握。Bootstrap 具有如下优势。

（1）响应式设计：Bootstrap 具有内置的响应式栅格系统，可以根据不同设备的屏幕大小自动调整布局和组件的显示方式。

（2）移动优先：Bootstrap 采用了移动优先的设计理念，使得开发的网页在移动设备上具有良好的用户体验，同时兼容桌面和平板设备。

（3）CSS 组件：Bootstrap 提供了丰富的可重用 CSS 组件，如按钮、导航栏、表单、卡片、警告框等，使开发人员能够快速构建功能丰富的用户界面。

（4）JavaScript 插件：Bootstrap 还包含一些常用的 JavaScript 插件，如轮播图、下拉菜单、模态框、滚动监听等，提供了丰富的交互功能。

（5）样式定制：Bootstrap 提供了多个主题和样式变量，使开发人员能够根据自己的需求进行定制，以满足不同项目的设计风格。

（6）文档和社区支持：Bootstrap 拥有详细的官方文档，展示了组件的使用方法和示例代码，并提供了一系列的样板文件和模板，帮助开发人员快速入门。此外，Bootstrap 拥有庞大的社区支持，有许多开发人员共享了自己的经验和资源。

2. Bootstrap 框架的下载与安装

Bootstrap 的使用方式有两种：一种是离线，另一种是在线引用。

（1）离线使用

①登录 Bootstrap 中文官网（<https://www.bootcss.com>），如图 4-30 所示。



图 4-30 Bootstrap 首页

②选择所需版本，单击对应的“Bootstrap v4 中文文档”按钮，进入主页 <https://v4.bootcss.com>，如图 4-31 所示。



图 4-31 Bootstrap 下载主页

③单击“下载 Bootstrap”按钮，进入下载页面，如图 4-32 所示。



图 4-32 Bootstrap 离线使用

④单击图中“下载 Bootstrap”按钮下载，将下载好的压缩包文件存放至 Web 项目根文档。bootstrap-3.4.1-dist.zip 解压后重命名为 bootstrmp，且应包括以下文件夹。

- CSS 文件夹：CSS 核心文件和主题文件。
- JS 文件夹：JavaScript 核心文件。
- fonts 文件夹：字体图标文件。

⑤Bootstrap 的使用依赖 jQuery，进入 jQuery 官网 (<https://jquery.com>) 下载 jQuery，如图 4-33 所示。

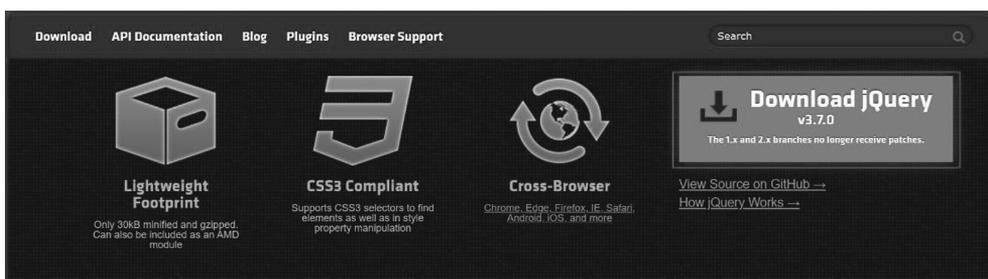


图 4-33 jQuery 官网主页

⑥点击“Download jQuery”按钮进入下载页面。建议初学者选择未压缩版，点击“Download uncompressed, development jQuery 3.7.0”，展开 jquery-3.7.0.js 源码，全选 jquery-3.7.0.js 源码（快捷键 Ctrl+A），再复制（快捷键 Ctrl+C）。在 Web 项目的 bootstrap 文件夹下新建空文件 jquery.js，将复制内容粘贴到该文件内。index.html 引入 bootstrap 的部分代码：

```
<head>
...
<!-- 将下载好的 bootstrap-3.4.1-dist 复制到项目根文档 -->
<link rel="stylesheet" href="bootstrap/css/bootstrap.min.css">
<!-- 加载 bootstrap 框架依赖的 jQuery 文件 (bootstrap.js 之前) -->
  <script src="bootstrap/js/jquery.js"></script>
  <!-- 加载 Bootstrap 的所有 JS 插件, 依赖 jQuery, 放后 -->
  <script src="bootstrap/js/bootstrap.js"></script>
</head>
```

(2) 在线引用方式

使用 BootCDN，无须下载任何内容，直接单击“Copy”按钮复制链接到 HTML 文档即可使用，如图 4-34 所示。



图 4-34 BootCDN 方式

index.html 引入 bootstrap 的部分代码如下：

```
<head>
...
<!-- CSS -->
<link href="https://cdn.bootcdn.net/ajax/libs/twitter-bootstrap/4.6.2/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-xOol-HFLEh07PJGoPkLv1IbcEPTntaed2xpHsD9ESMhqIYd0nLMwNLD69Npy4HI + N" crossorigin="anonymous">
<!-- jQuery and JavaScript Bundle with Popper -->
<script src="https://cdn.bootcdn.net/ajax/libs/jquery/3.5.1/jquery.slim.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OG-pamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>
<script src="https://cdn.bootcdn.net/ajax/libs/twitter-bootstrap/4.6.2/js/bootstrap.bundle.min.js" integrity="sha384-7ymO4nGrkm372 HoS-bq1OY2DP4pEZnMiA+E0F3zPr+JQQtQ82gQ1HPY3QIVtztVua" crossorigin="anonymous"></script>
</head>
```

3. Bootstrap 的简单应用

打开 Bootstrap 官网，单击“快速入门”按钮，进入中文文档页，单击左侧导航栏“组件”进入 Bootstrap 全局样式页面，如图 4-35~图 4-37 所示。



图 4-35 Bootstrap v4 中文版页面



图 4-36 中文文档页



图 4-37 Bootstrap 全局样式页面

确保使用 HTML5 文档类型 (doctype) 并包含一个设置 viewport (视口) 的 <meta> 标签, 以实现正确的响应式布局。如果没有这一设置, 就会看到一些古怪的、不完整的样式。接下来, 根据项目所需进入各个组件开始 Bootstrap 之旅吧。

案例 4-15: 导航栏的简单应用。

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>bootstrap 实现导航栏</title>
```

```
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-
scale=1">
<!-- CDN 方式引入 bootstrap -->
<link href="https://cdn.jsdelivr.net/npm/bootstrap@ 5.3.0-al-
pha1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-
GLh1TQ8iRABdZL1603oVMWSktQOp6b7In1Zl3/Jr59b6EGGoI1aFkw7cmDA6j6gD
"crossorigin="anonymous">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@ 5.3.0-al-
pha1/dist/js/bootstrap.bundle.min.js" integrity=" sha384 -/mhDoLbDld
Zc3qpsJHpLogda//BVZbgYuw6kof4u2FrCedxOtgRZDTHgHUhOCVim " crossorigin
="anonymous"></script>
</head>
<body>
<nav class="navbar navbar-expand-sm bg-primary navbar-dark">
  <div class="container-fluid">
    <a class="navbar-brand">购物天堂</a>
    <div class="collapse navbar-collapse">
      <ul class="navbar-nav me-auto">
        <li class="nav-item">
          <a class="nav-link">热销商品</a>
        </li>
        <li class="nav-item">
          <a class="nav-link">电器</a>
        </li>
        <li class="nav-item">
          <a class="nav-link">手机</a>
        </li>
        <li class="nav-item">
          <a class="nav-link">电脑</a>
        </li>
        <li class="nav-item">
          <button class="btn btn-success btn-close-white"
type="button">登录/注册</button>
        </li>
      </ul>
    </div>
  </div>
</nav>
```

```
<form class="d-flex">
  <input class="form-control me-sm-2" type="text"
placeholder="输入商品搜索">
  <button class="btn btn-success">
    <svg xmlns="http://www.w3.org/2000/svg" width="16"
height="16" fill="currentColor" class="bi bi-search" viewBox="0 0 16 16">
      <path d="M11.742 10.344a6.5 6.5 0 1 0-1.397 1.398h-
.001c.03.04.062.078.098.115l3.85a1 1 0 0 1.415-1.414l-3.85-3.85a1.007
1.007 0 0 0-.115-.115z" data-bbox="11 11 11 11"/>
    </svg>
  </button>
</form>
</div>
</div>
</nav>
</body>
</html>
```

显示效果如图 4-38 所示。



图 4-38 导航栏简单应用

Bootstrap 提供了丰富的组件和功能，帮助开发人员能够快速、高效地构建现代化、具有响应式设计的网页。无论是对于初学者，还是有经验的开发人员，Bootstrap 都是一个方便实用的工具。

课后习题

一、单选题

1. 在 CSS3 中，用于选择没有子元素或文本内容为空的选择器是（ ）。
A. :not
B. :empty
C. :root
D. :after
2. 关于 CSS3 新增属性，下列说法正确的是（ ）。
A. 新增了多重背景属性
B. 支持圆角边框
C. 提供了动画、变形、多列等
D. 以上都正确

