

任务 2 编写网页模板

2.1 任务导入

网页信息延续了传统的文档、新闻报纸、表单等信息结构,比如一篇新闻总是有标题、作者、发稿日期、新闻内容和新闻图片内容构成,网页只不过是这些传统信息的电子化和多媒体化呈现,用户的阅读习惯并未彻底颠覆。网页模板代表了一般网页页面基本构成,主要包括 HTML 文档定义、图文信息插入、样式表和 js 脚本引用,同时涉及编码(HTML、CSS 和 JavaScript)的规范及整合方式,可以说,编写网页模板是前端开发基本功,更是前端学习的起点。本任务制作一个京东页面,如图 2-1 京东页面效果图所示。

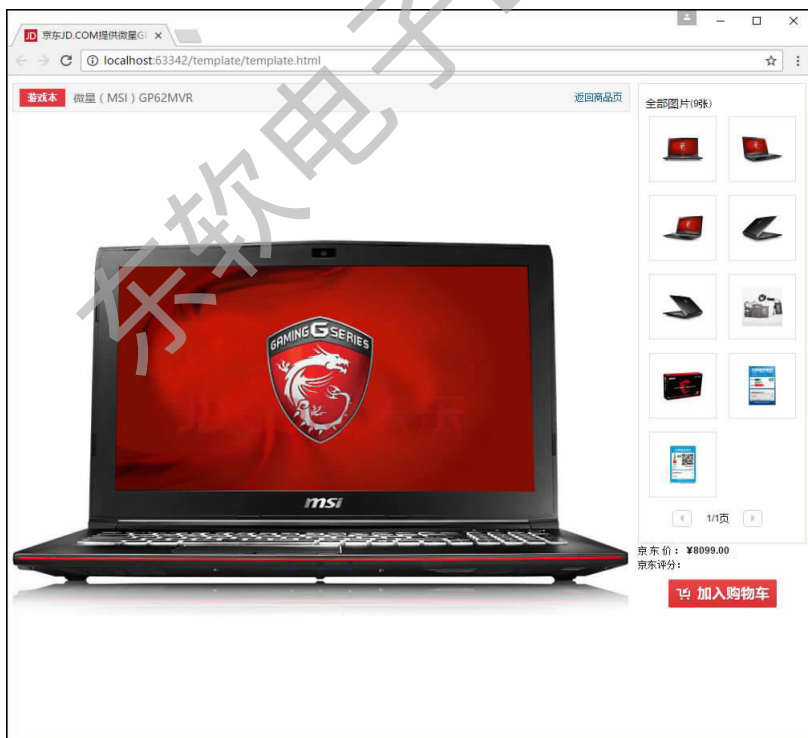


图 2-1 京东页面效果图

2.2 任务目标

技能目标	知识目标
1. 遵循规范编写 HTML、CSS 和 JS	1. Web 标准
2. 建立项目结构	2. Web 项目结构
3. 编写 HTML 文件模板	3. 通用规范
4. div 元素嵌套	4. HTML 规范
5. 无序列表使用	5. CSS 命名规范
6. 插入图像	6. JS 规范
7. 制作超链接	7. 图像规范
8. CSS 重置	8. DOCTYPE
9. 文字样式定义	9. meta
10. 浮动定位	10. favicon
11. HTML 链接到外部样式表	11. CSS 概述
12. document.getElementById()	12. CSS 语法
13. getElementsByTagName()	13. CSS 存在方式:链接式、内嵌式、行内、导入式
14. 循环遍历绑定事件 onclick 事件	14. CSS 继承与优先级规则
15. js 改变元素样式	15. CSS3 新增功能
16. HTML 链接外部 JS 脚本	16. CSS3 选择器类型
17. 浏览器测试	17. 链接 JS
	18. HTML 页面模板

2.3 核心知识

2.3.1 Web 标准

W3C(World Wide Web Consortium)指万维网联盟,创建于1994年10月,其宗旨是促进通用协议的发展,W3C负责创建并维护WWW标准。W3C的会员包括了软件开发商、内容提供商、企业用户、通信公司、研究机构、研究实验室、标准化团体以及政府,目前会员有IBM、Microsoft、America Online、Apple、Adobe、Macromedia、Sun Microsystems。

Web标准不是某一个标准,而是一系列标准的集合。网页主要由三部分组成:结构(Structure)、表现(Presentation)和行为(Behavior)。对应的标准也分三方面:结构化标准语言HTML、XML和XHTML,表现标准语言CSS,行为标准对象模型DOM、JavaScript等。遵循Web标准的页面,文档结构要求由HTML+CSS+JavaScript分而治之,如图2-2 Web标准构成所示。

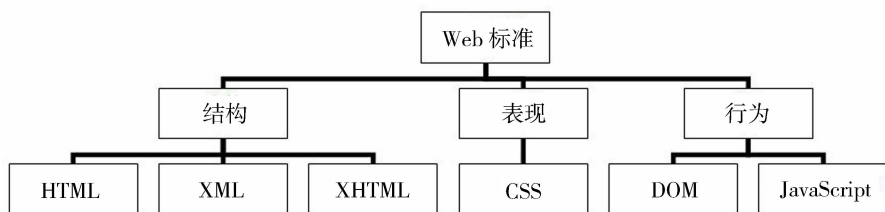


图 2-2 Web 标准构成

2.3.2 Web 项目结构

对于规模较大的项目,应将不同类别的页面保存在不同的文件夹中,形成了层次清晰的网站项目结构,把所有文件塞入到一个目录将变得难以管理。建立 Web 项目结构的原则是以最少的层次提供最清晰简洁的文件组织结构。在构建网站项目目录结构时,不要使用中文的目录名,不要使用过长的目录名,尽量使用有意义的目录名,使目录名跟内容相关,做到见名知意;目录的层次不要太多,建议不要超过 3 层;可以为每个主页创建一个主目录,而在这个主目录下还有子目录,子目录的命名最好跟栏目内容相关;对于需要经常更新的栏目可以建立独立的子目录,而一些相关性强,不需要经常更新的栏目,可以合并放在一个统一的目录中。具体前端开发实践中的建议是:项目名称用项目对应的英文关键词命名;文件及文件夹全部英文小写字母(如 login),可以使用中线(如 my-order),不可出现其他字符;网站根目录一般只存放首页文件(如 index.html);根目录下的 images 文件夹用于存放各页面引用的公用图片,根目录下的 js 文件夹存放 JavaScript 等脚本,根目录下 lib 文件夹中文件需包含版本号(如 jquery-3.2.0.js),压缩文件需包含“min”关键词(如 jquery-3.2.0.min.js),其他插件则可不包含,根目录下 CSS 文件夹存放样式表文件。根目录下的 media 文件夹下存放 flash、mp4、wav 等多媒体文件。多语言版本 lang 文件夹存放各语言版本。例如:简体中文存放在 cn 目录下,繁体中文存放在 tw 目录下,英文存放在 en 目录下等。

2.3.3 通用规范

1. 省略协议部分

protocol-relative URL 方案推荐省略图片、样式、脚本以及其他媒体文件 URL 的协议部分(http:和https:),好处是无论使用 HTTPS 还是 HTTP 访问页面,浏览器都会以相同的协议请求页面中的资源,同时可以节省一部分字节。除非文件在两种协议下都不可用(例如,直接从资源管理器打开网页时仍需加http:或者https:)

```

<! --不推荐 -->
<script src="https://www.google.com/js/gweb/analytics/autotrack.js"></script>
<! --推荐 -->
<script src="//www.google.com/js/gweb/analytics/autotrack.js"></script>
/* 不推荐 */
.example {background: url("https://www.google.com/images/example");}
/* 推荐 */
.example { background: url("//www.google.com/images/example");}
  
```

2. 缩进

一次缩进 2 个空格,不要使用 tab 或者混合 tab 和空格的缩进。

```
<ul>
  <li>Fantastic
  <li>Great
</ul>
.example {
  color: blue;
}
```

3. 大小写

HTML 元素名称、属性、属性值(除非 text/CDATA)、CSS 选择器、属性和属性值统一小写。

```
<! --不推荐 -->
<A HREF = "/">Home</A>
<! --推荐 -->

/* 不推荐 */
color: #E5E5E5;
/* 推荐 */
color: #e5e5e5;
结尾空格
<! --不推荐 -->
<p>What? _
<! --推荐 -->
<p>Yes please.
```

4. 编码

在 HTML 中通过 `<meta charset="utf-8">` 指定编码方式,CSS 中默认是 UTF-8,不需要指定。

5. 注释

注释用来解释代码,说明代码包含的模块、功能以及优点,注释写法如下:

```
<! -- HTML 注释-->
/* CSS 注释 */
// JavaScript 注释
```

6. 任务项

用 TODO 来标记待办事项,而不是用一些其他的标记,比如@@。

```
<! -- TODO: remove optional tags -->
<ul>
  <li>Apples</li>
  <li>Oranges</li>
</ul>
```

2.3.4 HTML 规范

HTML(HyperText Markup Language,超文本标记语言)代码由包含在尖括号(< >)中的 HTML 元素构成,元素通常由起始标签和结束标签(结束标签多一个斜杠/)两个标签组成,标签的作用就像是容器,用来把信息包裹起来并定义信息呈现的先后位置。元素属性出现在开始标签中,由属性名称和属性值构成,中间用等号(=)隔开,元素属性为元素设置了附加信息,只有少量的属性(如 lang、class、title、style)可以使用在所有标签中,大部分属性只能用在特定的标签中,比如 href 属性。

(1)文档类型。HTML 文档应使用 HTML5 的文档类型:<! DOCTYPE html>。

(2)孤立标签无需封闭自身。
不要写成
。

(3)HTML 正确性。尽可能使用正确规范的 HTML 结构和标签。

```
<! --不推荐 -->
<title>Test</title>
<article>This is only a test.
<! --推荐 -->
<! DOCTYPE html>
<meta charset="utf-8">
<title>Test</title>
<body>
<article>This is only a test.</article>
</body>
```

(4)语义化。

```
<! --不推荐 -->
<div onclick="goToRecommendations();">All recommendations</div>
<! --推荐 -->
<a href="recommendations/">All recommendations</a>
```

(5)多媒体元素降级。对于图片、视频、canvas 动画等多媒体元素,确保提供其他可访问的内容,图片可以使用替代文本(alt),视频和音频可以使用文字版本。

```
<! --不推荐 -->

<! --推荐 -->

```

(6)关注分离。标记、样式和脚本分离,确保相互耦合最小化,分而治之。

(7)实体引用。如果团队中文件和编辑器使用同样的编码方式,就没必要使用实体引用,如 —, ”, &# x263a;, 除了一些在 HTML 中有特殊含义的字符(如<和 &.)以及不可见的字符(如空格),小于号“<”编码为“<”,大于号“>”编码为“>”。

```

<! --不推荐 -->
The currency symbol for the Euro is &ldquo;&eur;&rdquo;,.
<! --推荐 -->
The currency symbol for the Euro is “”.
```

(8) type 属性。在引用样式表和脚本时,不要指定 type 属性,除非不是 CSS 或 JavaScript。因为 HTML5 中已经默认指定样式变的 type 是 text/css,脚本的 type 是 text/JavaScript。

```

<! --不推荐 -->
<link rel="stylesheet" href="//www.google.com/css/maia.css"
  type="text/css">
<! --推荐 -->
<link rel="stylesheet" href="//www.google.com/css/maia.css">
<! --不推荐 -->
<script src="//www.google.com/js/gweb/analytics/autotrack.js"
  type="text/javascript"></script>
<! --推荐 -->
<script src="//www.google.com/js/gweb/analytics/autotrack.js"></script>
```

(9)HTML 属性值用双引号。

```

<! --不推荐 -->
<a class='maia-button maia-button-secondary'>Sign in</a>
<! --推荐 -->
<a class="maia-button maia-button-secondary">Sign in</a>
```

(10)尽量减少标签层级,实现扁平化 DOM 结构。

(11)HTML 元素及属性名称小写,所有的属性必须被赋值,没有值的属性就用自身来赋值,属性值必须用英文引号" "括起来;

(12)避免使用已过时标签,如:"","<frame>","<s>"等。

(13)标签默认缺省格式:。

(14)<a>标签缺省格式:xxx,target 属性根据需求决定。

(15)HTML 注释格式为<! --注释内容-->,不要在注释正文中使用"--",它只能出现在 XHTML 注释的开头和结束。

(16)用 id 属性代替 name 属性。

2.3.5 CSS 命名规范

(1)id 和 class 命名。使用有含义明确的 id 和 class 名称。

```

/* 不推荐: 含义不明确 */
#yee-1901 {}
/* 不推荐: 按直觉来的 */
.button-green {}
.clear {}
/* 推荐: 指定含义 */
#gallery {}
#login {}
.video {}
/* 推荐: 通用 */
.aux {}
.alt {}

```

(2) id 和 class 命名风格。id 和 class 应该尽量简短,同时要容易理解。

```

/* 不推荐 */
#navigation {}
.atr {}
/* 推荐 */
#nav {}
.author {}

```

(3) 选择器。除非必要,否则不要在 id 或 class 前加元素名。

```

/* 不推荐 */
ul#example {}
div.error {}
/* 推荐 */
#example {}
.error {}

```

(4) 属性简写。尽量使用 CSS 中可以简写的属性(如 font, border, margin, padding, background),可以提高编码效率以及代码可读性。

```

/* 不推荐 */
border-top-style: none;
font-family: palatino, georgia, serif;
font-size: 100%;
line-height: 1.6;
padding-bottom: 2em;
padding-left: 1em;
padding-right: 1em;
padding-top: 0;
/* 推荐 */
border-top: 0;
font: 100% /1.6 palatino, georgia, serif;
padding: 0 1em 2em;

```

(5) 0 和单位。值为 0 时不用添加单位,如 `margin:0`。属性值开头的 0 或值在 -1 和 1 之间时,不需要加 0,如 `font-size:.8em`。

(6) 16 进制表示法。

```
/* 不推荐 */
color: #eebbcc;
/* 推荐 */
color: #ebc;
```

(7) 前缀。使用带前缀的命名空间可以防止命名冲突,同时提高代码可维护性。

```
.adw-help {} /* AdWords */
#maia-note {} /* Maia */
```

(8) id 和 class 命名分隔符。选择器中使用连字符可以提高可读性。

```
/* 不推荐: “demo” 和 “image” 之间没有分隔符 */
.demoimage {}
/* 不推荐: 使用下划线 */
.error_status {}
/* 推荐 */
#video-id {}
.ads-sample {}
```

(9) CSS 书写顺序。按照属性首字母顺序书写 CSS 易于阅读和维护,排序时忽略带有浏览器前缀的属性。

```
background: fuchsia;
border: 1px solid;
-moz-border-radius: 4px;
-webkit-border-radius: 4px;
border-radius: 4px;
color: black;
text-align: center;
text-indent: 2em;
```

(10) 块级内容缩进。为了反映层级关系和提高可读性,块级内容都应缩进。

```
@media screen, projection {
  html {
    background: #fff;
    color: #444;
  }
}
声明结束,每行 CSS 都应以分号结尾。
/* 不推荐 */
```



```
.test {
  display: block;
  height: 100px
}
/* 推荐 */
.test {
  display: block;
  height: 100px;
}
```

(11) 属性名结尾。属性名和值之间都应有一个空格。

```
/* 不推荐 */
h3 {
  font-weight:bold;
}
/* 推荐 */
h3 {
  font-weight: bold;
}
```

(12) 声明样式块的分隔。在选择器和{}之间用空格隔开。

```
/* 不推荐: 缺少空格 */
#video{
  margin-top: 1em;
}
/* 不推荐,增加了多余行 */
#video
{
  margin-top: 1em;
}
/* 推荐 */
#video {
  margin-top: 1em;
}
```

(13) 选择器分隔。每个选择器都另起一行。

```
/* 不推荐 */
a:focus, a:active {
  position: relative; top: 1px;
}
/* 推荐 */
h1,
h2,
h3 {
  font-weight: normal;
  line-height: 1.2;
}
```

(14)规则分隔。规则之间都用空行隔开。

```
html {  
    background: # fff;  
}  
  
body {  
    margin: auto;  
    width: 50 % ;  
}
```

(15)CSS 引号。属性选择器和属性值用单引号,URI 的值不需要引号。

```
/* 不推荐 */  
@import url("//www.google.com/css/maia.css");  
html {  
    font-family: "open sans", arial, sans-serif;  
}  
/* 推荐 */  
@import url(//www.google.com/css/maia.css);  
html {  
    font-family: 'open sans', arial, sans-serif;  
}
```

(16)分段注释。用注释把 CSS 分成各个部分。

```
/* Header */  
  
# adw-header {}  
  
/* Footer */  
  
# adw-footer {}  
  
/* Gallery */  
  
. adw-gallery {}
```

(17)样式文件命名规范。全局样式 reset. css, 框架布局 layout. css, 外观样式 base. css, 页面样式 page. css(页面特有样式,可选), 打印样式 print. css(可选)。

(18)所有的命名用小写的英文单词。

(19)单词语义明确,不使用简单的方位词直接命名,如 left, CSS 命名参考见表格 2-1 所示。

表格 2-1

CSS 命名参考

CSS 样式命名	说明
wrapper	页面外围控制整体布局宽度
container 或 content	容器,用于最外层
layout	布局
head, header	页头部分
foot, footer	页脚部分
nav	主导航
sub_nav	二级导航
menu	菜单
sub_menu	子菜单
side_bar	侧栏
sidebar_l, sidebar_r	左边栏或右边栏
main	页面主体
tag	标签
msg message	提示信息
tips	小技巧
vote	投票
friendlink	友情链接
title	标题
summary	摘要
login_bar	登录条
search_input	搜索输入框
hot	热门热点
search	搜索
search_output	搜索输出和搜索结果相似
search_bar	搜索条
search_results	搜索结果
copyright	版权信息
branding	商标
logo	网站 LOGO 标志
site_info	网站信息
site_info_legal	法律声明

(续表)

CSS 样式命名	说明
site_info_credits	信誉
join_us	加入我们
partner	合作伙伴
service	服务
regsiter	注册
arr_arrow	箭头
guild	指南
site_map	网站地图
list	列表
home_page	首页
sub_page	二级页面子页面
tool,toolbar	工具条
drop	下拉
dorp_menu	下拉菜单
status	状态
scroll	滚动
tab	标签页
left right center	居左、中、右
news	新闻
download	下载
banner	广告条(顶部广告条)

(20)不缩写单词,除非一看就明白的单词,例如 btn、msg。

(21)长名称或词组可以使用下划线作为连接符,将每个单词用中划线或下划线分隔,例如,drop_meun、sub_nav_menu。

(22)选择器嵌套层级不超过 3 级。

(23)id 应该按需使用,而不能滥用。

(24)使用 CSS 缩写属性,比如 padding:0 10px 5px 5px 等,这样精简代码同时又能提高用户的阅读体验。

(25)向“无 id,无层级,无标签”准则靠拢,可有效提高重用性,减小文件大小,提升渲染效率。

(26)注释。区分页面各部分的注释,如/*****产品中心*****/。

(27)模块的注释,如/* 首页导航栏 */。

(28)CSS 各属性的排列顺序:Positioning(定位,如 position, top, z-index)→Box model (盒模型,如 display, box-sizing, width, border)→Typographic(排版,如 font, line-height,

text-align)→Visual(视觉,如 background, color, opacity)→Other(其他,如 cursor),定位(positioning)可以从正常的文档流中移除元素,并且还能覆盖盒模型(box model)相关的样式,因此排在首位。盒模型决定了组件的尺寸和位置,因此排在第二位。其他属性只是影响组件的内部(inside)或者是不影响前两组属性,因此排在后面。

(29)CSS 格式化。使用不换行方式书写,增加书写速度,. box{background:none repeat scroll 0 0 transparent;bottom:11px;position:relative;width:22px;z-index:33;}。

(30)CSS 字体单位。px 像素(pixel)是相对于显示器屏幕分辨率而言;em 是相对于当前对象内文本的字体尺寸。当前对行内文本的字体尺寸未被人为设置,则相对于浏览器的默认字体尺寸;rem 相对于 HTML 根元素。

2.3.6 JS 规范

(1)文件命名。使用短线(-)或句点(.)作为分隔符号,推荐使用句点,最好使用小写英文字符,不要使用其他符号和扩展字符,如 jQuery UI 1.9.0 的源文件可命名为“jquery-ui-1.9.0.js”,使用.js 扩展名,这个扩展名的兼容性最好。用句点分隔表示名称中的从属关系,范围大的在前,如 jQuery 的表单插件 Form 既可以命名为“jquery.form.js”。

(2)变量命名。命名方法采用小驼峰式命名,变量前缀应用名词,尽量在变量名字中体现所属类型,如:length、count 等表示数字类型,name、title 表示为字符串类型。示例:

```
//好的命名方式
var maxCount = 10;
var tableTitle = 'LoginTable';
//不好的命名方式
var setCount = 10;
var getTitle = 'LoginTable';
```

(3)函数命名。命名方法采用小驼峰式命名法,函数名字前缀用动词,以此区分变量和函数,可使用常见动词约定如表格 2-2 所示。

表格 2-2

函数命名约定动词

动词	含义	返回值
can	判断执行动作(权限)	返回一个布尔值,true 可执行:false 不可执行
has	判断是否含有某个值	返回一个布尔值,true 含有此值:false 不含此值
is	判断是否为某个值	返回一个布尔值,true 为某个值:false 不为某个值
get	获取某个值	返回一个非布尔值
set	设置某个值	无返回值、返回是否设置成功或者返回链式对象
load	加载某些数据	无返回值或者返回是否加载完成的结果

(4)常量命名命名方法。名称全部大写,使用大写字母和下划线来组合命名,下划线用以分割单词,如 MAX_COUNT,URL。

(5)注释格式。使用多行注释,以/* 开头,* /结尾。

- (6)换行缩进采用 4 个空格。
- (7)结束行需添加分号“;”。
- (8)尽量选用局部变量而不是全局变量。
- (9)尽量使用链式写法。
- (10)尽量减少 DOM 调用。
- (11)将 js 脚本放到页面底部。
- (12)使用 jquery 的 data 来存取数据,减少对 dom 的操作。
- (13)jquery 的推荐使用 on 方法绑定事件。
- (14)尽量不用标签选择器,能用 id 选择器的就不用 class 选择器。

2.3.7 图像规范

- (1)图像处理或者切图时保存 Web 格式(jpg、png、gif 或者 webp),减小图片大小;
- (2)图片尺寸一律采用整数,如 20×20,50×50;
- (3)小图片一律要合并,如 Sprite(雪碧图),并保存对应的 psd 源文件,以便后期修改。

2.3.8 DOCTYPE(文档类型)

由于 HTML 存在多个版本,因此每个网页的开头都应该用一个 DOCTYPE 声明此页使用的 HTML 版本,以帮助浏览器正确地渲染页面。<! DOCTYPE>声明不是 HTML 标签,没有结束标签,对大小写不敏感,必须是 HTML 文档的第一行,位于<html>标签之前,每个文档只能有一个<! DOCTYPE>,W3C 提供了一个验证文档有效性的免费在线服务 <http://validator.w3.org/>。HTML5 文档的开头位置使用<! DOCTYPE html>来申明,HTML5 不基于 SGML,所以不需要引用 DTD。

2.3.9 meta

meta 元素可提供有关某个 HTML 文档的元信息(meta-information),比如描述、针对搜索引擎的关键词以及刷新频率,如凤凰网的 meta 定义为:

```
<meta charset="UTF-8">
<title>凤凰网</title>
<meta content="凤凰,凤凰网,凤凰新媒体,凤凰卫视,凤凰卫视中文台,资讯台,电影台,欧洲台,美洲台,凤凰周刊,phoenix,phoenixtv" name="keywords" />
<meta name="description" content="凤凰网是中国领先的综合门户网站,提供含文图音视频的全方位综合新闻资讯、深度访谈、观点评论、财经产品、互动应用、分享社区等服务,同时与凤凰无线、凤凰宽频形成三屏联动,为全球主流华人提供互联网、无线通信、电视网三网融合无缝衔接的新媒体优质体验。" />
<meta content="index,follow" name="robots" />
<meta content="凤凰网" name="author" />
<meta property="qc:admins" content="17512667376165676375" />
<meta content="Copyright 1999-2014. www.ifeng.com . All Rights Reserved." name="copyright" />
```

```

<meta name="application-name" content="凤凰网" />
<meta name="msapplication-TileColor" content="#8F1C21" />
< meta name = " msapplication-TileImage " content = " http://p2. ifengimg. com/
8cbe73a7378dafdb/2013/0416/logo.png" />

```

charset 属性规定 HTML 文档的字符编码。charset 属性可以通过任意元素上的 lang 属性来重写。HTML5 meta 元素的 charset 语法 `<meta charset="character_set">`, character_set 规定 HTML 文档的字符编码。常用值为 Unicode 字符编码 UTF-8, 详细情况请了解 IANA 字符集 (<http://www.iana.org/assignments/character-sets/character-sets.xhtml>)。

keywords 用于包含一组以逗号分隔的关键词列表, 用户可能通过关键词来搜索到页面, 对搜索引擎建立网页的索引有参考意义。keywords 出现在 name 属性中, 使用 content 属性提供网页的关键词。

description 提供的网页简短描述, 支持搜索引擎对网页进行分析, 网页简短描述不能太长, 要跟网页内容密切相关, 网页制作者滥用 description(提供与网页无关的简短描述), 导致目前常用的搜索引擎降低了 description 的重要性。description 出现在 name 属性中, 使用 content 属性提供网页的简短描述。

robots 用于指定搜索机器人哪些页面需要索引, 哪些页面不需要索引。content 的参数有 all、none、index、noindex、follow、nofollow。默认是 all, 文件将被检索, 且页面上的链接可以被查询; none 文件将不被检索, 且页面上的链接不可以被查询; index 文件将被检索; follow 页面上的链接可以被查询; noindex 文件将不被检索, 但页面上的链接可以被查询; nofollow 文件将不被检索, 页面上的链接可以被查询。

author 用于定义网页的设计者, copyright 申明页面版权, pragma 用于防止浏览器对页面的缓存, expires 用于指定页面的过期时间, 即缓存的有效期。application-name 元数据定义固定网站应用程序实例的名称。

2.3.10 favicon

favicon 收藏夹图标出现在浏览器地址栏左侧的那个小图标, 根据浏览器的不同, 还会同时出现在标签式浏览器的标签卡上。favicon 文件生成可使用专门的收藏夹图标编辑器制作。在 HTML 中使用 link 标签必须在 head 标签里(在 `<head>` 和 `</head>` 之间)来设置 favicon, 其中 href 属性指定网站图标路径和文件名称, type 属性设置 MIME 标识, ico 文件为 image/vnd.microsoft.icon(或者 image/x-icon)、gif 文件为 image/gif、png 文件为 image/png。网站图像通常可以使用任何被浏览器支持的图像格式, .ico 文件格式通常可以被所有可以显示 favicon 的浏览器读取, 通用性最强。ico 常用像素宽高是 16×16 像素、32×32 像素、48×48 像素, 位深多数使用 4 位、8 位、24 位。

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

2.3.11 CSS 概述

HTML 标记原本用于定义文档结构和内容,但随着浏览器开发商对 HTML 标签和属性的不断扩展,致使 HTML 文档内容和表现混杂在一起,大大降低了网页界面设计和内容更改的效率,为此 W3C 制定了 CSS(Cascading Style Sheets,层叠样式表),其目标用 CSS 取代 HTML 表格式布局设计、框架和其他表现的控制标记,与 HTML 网页结构和内容分离,实现内容与表现分离,使网站的构建及维护更加容易。使用 CSS 的好处有:

(1)提高网站制作效率。如果没使用 CSS 时,想要更新整个站点中所有文字样式和版式,就必须一页一页地修改,而当使用 CSS 链接到站点的所有网页时,仅更新 CSS 文档,就能同时改变站点中所有页面的布局和外观。

(2)CSS 能简化网页的格式代码,网页文件更小,网络传输更快,达到网页减肥效果。

(3)保持网页在大多数浏览器效果一致性。

(4)实现了内容和表现的分离。回归了 HTML 定义网页的结构,CSS 定义表现形式,提供了对布局和图文排版的精细控制。

2.3.12 CSS 语法

CSS 语法包括选择器、样式属性和属性值三部分,基本语法:

```
selector { property: value; property: value; ... property: value }
```

selector 选择器可以是 HTML 标记、id、class 和复合样式等,property 是样式属性,value 是样式属性值,样式属性和属性值用冒号(:)分开。定义多个属性时用分号(;)将各属性分开,最后一条样式属性和属性值也加分号。为提高代码的可读性,规则之间都用空行隔开,组选择器分行写,选择器和{}之间、属性名和值之间都输入一个空格,样式属性和属性值分行写,每行只定义一个样式属性,将 h1 元素内的文字颜色定义为红色,同时将字体大小设置为 14 像素。在这个例子中,h1 是选择器,color 和 font-size 是属性,red 和 14px 是属性值,如图 2-3 样式定义语法结构所示。

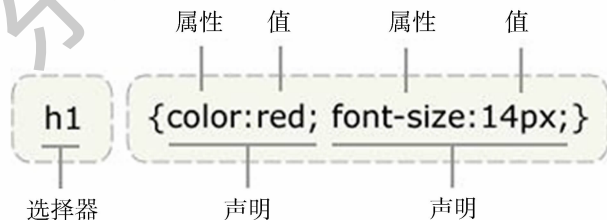


图 2-3 样式定义语法结构

属性值由多个单词组成,则必须给值加上引号,如 `p {font-family:'sans serif'};`。

相同属性和值的选择符组合起来称为选择符组,选择符间用“,”分隔,这样可以减少重复定义样式,如将所有的标题颜色定义为绿色的样式定义如下。

```
h1,
h3 {
    color: green;
}
```


2.3.13 CSS 存在方式

1. 链接式

链接式样式是使用频率最高,也最为实用的方法。链接式样式的最大优势在于 CSS 代码与 HTML 代码完全分离,并且同一个 CSS 文件可以被不同的 HTML 所链接使用。因此在设计整个网站时,可以将所有页面都链接到同一个 CSS 文件,使用相同的样式风格。如果整个网站需要进行样式的修改,就仅仅只需要修改这一个 CSS 文件就能改变整个站点的外观,可以大大减少网页的代码,提高网站制作和更新的效率。当需要引用外部样式文件时,在<head>标签通过<link>标签来链接外部样式表。在 HTML 文档中使用 link 元素定义页面包含的外部 CSS 文件,link 元素位于 head 元素中,是空元素,无结束标签。用法格式为:

```
<link href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" rel="stylesheet">
```

href 属性指定 CSS 文件路径(一般位于 CSS 或 styles)和文件名,type 指定页面所链接文档的类型,属性值为 text/css,rel 表明 HTML 页面与被链接文件的关系,链接到一个 CSS 文件时值为 stylesheet。

2. 内嵌式

内嵌样式表就是将 CSS 写在 HTML 文件的头部的<head>与</head>之间,并且以<style>开始,以</style>结束,基本语法:

```
<head>
  <style type="text/css">
    p {
      color: #FF00FF;
      font-size: 14px;
    }
  </style>
</head>
```

<style>标记用来说明所要定义的样式。type="text/css"声明是样式表代码,<!-->用于防止将样式代码显示在不支持样式的浏览器窗口中。

内部样式表将所有的样式表信息都列于 HTML 文件的头部的同一区域,方便后期的维护,页面本身也大大瘦身。但如果是一个网站,拥有很多的页面,对于不同页面的<p>标记都希望采用同样的风格时,内嵌式并非最优的选择,内嵌式仅适用于对单个网页需要特殊的样式时使用。

3. 行内样式

行内样式是所有样式方法中最为直接的一种,直接对 HTML 标记使用 style 属性,然后将 CSS 代码直接写在其中,基本语法:

```
<pstyle="font-size:12px;line-height:18px">文本段落</p>
```

利用这种方法定义的样式,其效果只能控制某个标记。只在一个网页一两个地方用到

的 CSS 才选用行内样式。由于要将表现和内容混杂在一起,内联样式会损失掉样式表的许多优势,请慎用、少用行内样式。

4. 导入式

采用 import 方式导入的样式表,在 HTML 文件初始化时,会被导入到 HTML 文件内,作为文件的一部分,类似于内嵌式的效果,而链接式样式表则是在 HTML 的标记需要格式时,才以链接的方式引入。导入样式最大的用处是可以让一个 HTML 文件导入很多的样式表。导入样式表在内部样式表的 <style> 里用 @import 导入一个外部样式表。@import 声明可以一般放在 <head> 标记内,基本语法:

```
<head>
  <style type="text/css">
    @import url("style.css");
  </style>
</head>
```

@import 语句后面的“;”不能省略,外部样式表文件的扩展名必须是 .css,样式表地址可以是绝对地址,也可以是相对地址。

2.3.14 CSS 继承与优先级规则

样式的继承可以减少对子元素样式的重复定义,像 font-family 这类可继承的样式会把属性继承给子元素。但 background-color、border 等属性不会被子元素继承。可以通过将属性值设置为 inherit 来强制大多数元素从它的父元素中继承属性值。

当网页同时使用了行内样式、内嵌式样式、链接样式和导入样式时,将会出现优先级的的问题。如果在各种方法中设置的属性不一样,例如内嵌式设置字体为宋体,链接式设置颜色为红色,那么显示结果将是红色的宋体字。但当各种方法同时设置一个属性时,就根据优先级确定最终样式。如果有两个或者更多的规则应用在同一元素上,将遵循以下优先级规则:

(1) 选择器都有一个权值,权值越大越优先。内联样式表的权值最高 1 000, id 选择器的权值为 100, class 类选择器的权值为 10, HTML 标签选择器的权值为 1, 复合样式的权值据上面选择器的权值从左向右相加得出。如 #nav. main li a{} 的的权值为:100+10+1+1=112。

(2) 就近原则。如果两个选择器完全相同,后出现的选择器优先级较高。一般而言,样式优先级从低到高的顺序是:导入式→内嵌样式表→外部样式表→行内样式,行内样式拥有最高的优先权,导入样式优先权最低,但如果外部样式放在内部样式的后面引入,则外部样式将覆盖内部样式。

(3) 具体性原则。如果一个选择器比其他选择器更加具体,具体的选择器优先于一般的选择器,如 h3 比 * 具体,h3 span 比 h3 具体。

(4) 重要性。给 css 属性值添加 ! important 来强调这条规则比应用于同一元素的其他规则更重要,优先级最大。

(5) 创作者的规则高于浏览者,即网页编写者设置的 CSS 样式的优先权高于浏览器所设置的样式。

(6) 继承的 CSS 样式不如后来指定的 CSS 样式。

2.3.15 CSS3 新增功能

CSS3 与之前的几个版本相比变化很大,部分属性还不能够被浏览器完美地支持,CSS3 的浏览器支持情况可以访问 <http://caniuse.com/> 获取。目前能被浏览器支持较为完美、更具实用性的新特性有:

(1)强大的 CSS3 选择器。CSS3 选择器增加属性选择器、伪类选择器等新类型,允许设计师通过选择器直接指定需要的 HTML 元素,而不需要在 HTML 中添加不必要的类名、id 等,使用 CSS3 选择器更易做到结构与表现分离,设计师能轻松地设计出简洁、轻量级的 Web 页面,并且能更好地维护和修改样式。

(2)图片式视觉效果更易实现。Web 中最常见的效果包括圆角、阴影、渐变背景、半透明、图片边框等,CSS3 之前这些视觉效果都是依赖于设计师制作图片或者 JavaScript 脚本来实现,CSS 直接提供了样式支持。

(3)增强的背景可控性。传统的 CSS 中的背景设置可控性较差,CSS3 不再局限于背景色、背景图像的运用,添加了 background-origin、background-clip、background-size 等新属性,实现背景效果更加快捷灵活。

(4)盒模型与布局变化。CSS3 中设计师可以直接通过 CSS3 来实现弹性盒子、分栏排版等布局排版功能。

(5)阴影效果。CSS3 提供了文本阴影(text-shadow)和盒子阴影(box-shadow)两种阴影模式,制作修改方便快捷。

(6)Web 字体和 Web Font 图标。CSS3 重新引入@font-face 来链接服务器上的字体文件实现浏览器端的非系统字体显示功能,嵌入的字体能变成浏览器的安全字体,不再担心用户没有这些字体而无法正常显示的问题,从此不再使用图片代替特殊字体。

(7)颜色与透明度。CSS3 颜色模块不再局限于 RGB 和十六进制两种模式,增加了 HSL、HSLA、RGBA 几种新的颜色模式,能轻松地改变任意一个元素的透明度,另外,还可以使用 opacity 属性来实现元素的透明度。

(8)元素的变换。CSS3 引进了一个变形属性,可以在 2D 或者 3D 空间里操作盒容器的位置和形状,例如旋转、扭曲、缩放或者移位。

(9)CSS3 过渡与动画交互效果。CSS3 的过渡(transition)特性能在 Web 制作中实现一些简单的动画效果,让某些效果变得更具流线型、平滑性,而 CSS3 动画(animation)特性能够实现更复杂的样式变化,以及一些交互效果,而不需要使用 Flash 或者是 JavaScript 脚本代码。

(10)媒体特性与 Responsive 布局。CSS3 的媒体特性可以实现一种响应式(Responsive)布局,使布局可以根据用户的显示终端或设备特征选择对应的样式文件,从而在不同的显示分辨率或设备下具有不同的布局渲染效果,特别是在移动端上的实现更是一种理想的做法。

2.3.16 CSS3 选择器类型

选择器是 CSS 重要部分之一,也是 CSS 的根基。CSS 选择器用于获取目标元素,使某个样式应用于特定的 HTML 元素。CSS3 选择器在常规选择器的基础上新增了属性选择器

和伪类选择器,可以帮助在 Web 开发中减少对 HTML 类名或 id 名的依赖,以及对 HTML 元素的结构依赖,使编写代码更加简单轻松。CSS 选择器可分为基本选择器、层次选择器、属性选择器、伪类选择器和伪元素选择器 5 大类,如图 2-4 CSS3 选择器导图。

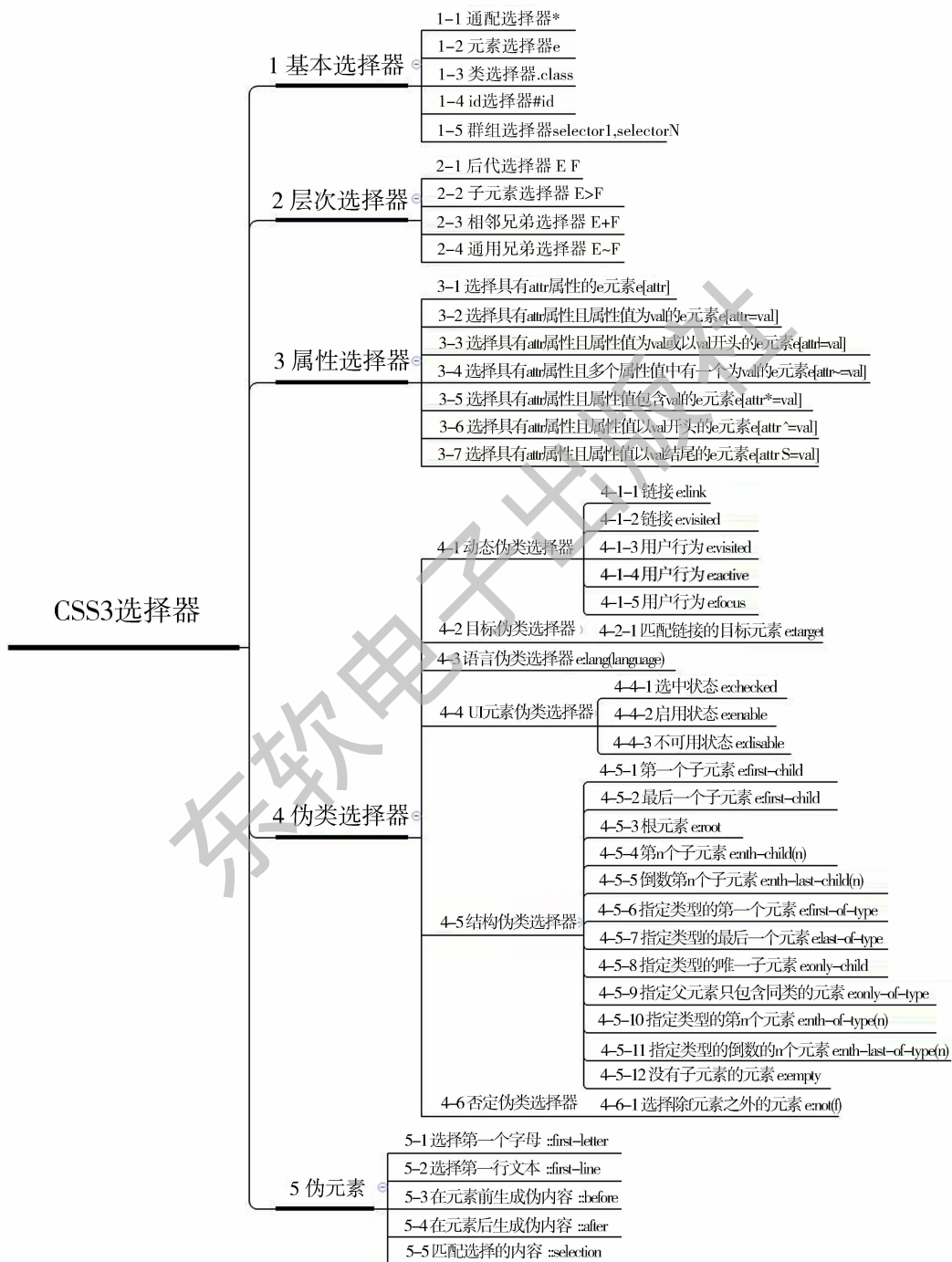


图 2-4 CSS3 选择器导图

CSS 常用选择器是基本选择器和层次选择器,基本选择器包括通配选择器、元素选择器、类选择器、id 选择器、群组选择器;层次选择器包括子元素选择器、后代选择器、相邻兄弟选择器、通用兄弟选择器,具体使用见表格 2-3 CSS 常用选择器所示。

表格 2-3 CSS 常用选择器

选择器	说明	举例
基本选择器		
通配选择器	应用于文档中所有元素	* {}
元素选择器	重新定义 HTML 元素	p {}
类选择器	选择元素的 class 值与 class 选择器.后面相同的元素	.wrap {}
id 选择器	选择元素的 id 值与 id 选择器#后面相同的元素	#header {}
群组选择器	选择选择器组中每个选择器匹配的元素	h1, p {}
层次选择器		
子元素选择器	选择指定元素的直接子元素	li>a {}
后代选择器	选择指定元素的后代元素(不限于指定元素的直接子元素)	li a {}
相邻兄弟选择器	选择一个元素的相邻的兄弟元素	h3+p {}
通用兄弟选择器	选择一个元素的兄弟元素,不要求相邻	h3~p {}

2.3.17 链接 JS

JavaScript 是一种解释型语言,不需要进行编译和构建。HTML 文档中包含 JavaScript 脚本有两种方式:直接在 HTML 页面中嵌入 JavaScript 代码和包含外部 JavaScript 脚本文件。尽管 2 种做法都可以,但采用外部文件具有以下优点:

(1)可维护性。确保 JavaScript 代码集中存放,不跟 HTML 代码混杂,编辑 JavaScript 代码更高效。

(2)可缓存。浏览器能够缓存外部 JavaScript 文件,两个页面使用同一个文件,只需下载一次,能够加快页面加载的速度。要特别注意的是,带有 src 属性的 script 元素不应该在其<script>和</script>标签之间再包含额外的 JavaScript 代码。如果包含了嵌入的代码,则只会下载并执行外部脚本文件,嵌入的代码会被忽略。在使用 script 元素嵌入 JavaScript 代码时,只须为 script 指定 type 属性并设置属性值为 text/javascript,如:

```
<script type="text/javascript">
  window.onload = function () {
    alert("Hi! JavaScript!")
  }
</script>
```

嵌入在 HTML 文档内部的 JavaScript 代码,不允许在代码的任何地方出现“</script>”字符串,因为浏览器会视</script>为<script>的结束标签,从而中断了代码的加载执行。此时,可通过转义字符“\”解决,即将</script>改写为<\/script>。

包含外部 JavaScript 脚本文件要通过使用 script 元素的 src 属性来指定被包含的外部

JavaScript 脚本文件,这些文件地址可以是本域名或者外域名的,引用外域名文件需确保文件安全可靠,外部脚本文件只须包含要放在 script 内部的那些 JavaScript 代码即可,本身不需要再包含 script 标签了,script 标签位置传统上放在 HTML 文档头部 head 元素内,目的是把所有外部文件(包括 CSS 文件和 JavaScript 文件)的引用都放在相同的地方。然而,在 head 元素中包含所有 JavaScript 文件,意味着必须等全部 JavaScript 代码都被下载、解析和执行完成以后,才能开始呈现页面的内容(浏览器在遇到 body 标签时才开始呈现内容)。对于包含了很多 JavaScript 代码的页面来说,无疑会导致浏览器显示页面延迟,而延迟期间的浏览器窗口将是一片空白。为避免这个问题,现代 Web 应用程序一般把全部 JavaScript 引用放在 body 元素结束标签之前,这样解析包含 JavaScript 代码之前,页面的内容将完全呈现在浏览器中,而用户也会因为浏览器显示空白页面的时间缩短而感到打开页面速度加快了。如:

```
<script src="https://code.jquery.com/jquery-3.2.0.min.js"></script>
```

跟解析嵌入式 JavaScript 代码一样,在解析外部 JavaScript 文件(包括下载该文件)时,页面处理会暂时停止,<script>元素,该元素定义了 6 个属性,见表格 2-4 script 元素属性:

表格 2-4 script 元素属性

属性	属性值	描述
async	async	可选,立即下载,但不妨碍页面中的其他操作(异步执行脚本),比如下载其他资源或等待加载其他脚本。仅适用于外部脚本文件。
charset	charset	可选,规定 src 属性值指定的脚本使用的字符编码,大多数浏览器会忽略它的值。仅适用于外部脚本文件。
defer	defer	可选,指定脚本可以延迟到文档完全被解析和显示之后再执行。仅适用于外部脚本文件。
src	URL	可选,指定外部脚本的 URL。
type	MIME-type	可选,指定编写代码使用的脚本语言的内容类型(也称为 MIME 类型),一般为 text/javascript(默认值),HTML5 可以省略。

script 标签的 defer 属性表明脚本在执行时不会影响页面的渲染,脚本会被延迟到整个页面都解析完毕后再运行,即立即下载,但延迟执行。script 标签的 async 用于改变处理脚本的行为,告诉浏览器立即下载文件,不让页面等待两个脚本下载和执行,从而异步加载页面其他内容,异步脚本一定会在页面的 load 事件前执行。但与 defer 属性不同,标记为 async 的脚本并不保证按照指定它们的顺序先后执行,因此 HTML 包括多个脚本时要确保脚本之间无依赖关系。无论如何包含代码,只要不存在 defer 和 async 属性,浏览器都会按照<script>元素在页面中出现的先后顺序依次进行解析。

2.3.18 HTML 页面模板

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>HTML 模板</title>
```

```

    <link href="css/styles.css" rel="stylesheet">
</script src="js/scripts.js"></script>
</head>
<body>
</body>
</html>

```

2.4 任务实施

2.4.1 建立项目

启动 WebStorm, 参照 1.4.4 节, 在驱动器 E 盘“wwwroot”建立项目根文件夹中“template”, 并依次建立 template.html、js 和 css 文件夹, 建立项目结构如图 2-5 所示。

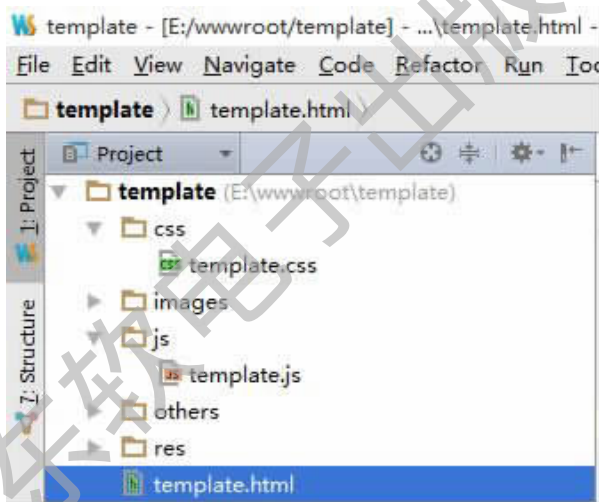


图 2-5 建立项目结构

2.4.2 编写 HTML 文件

双击打开项目根目录下的“template.html”, 参考 EMMET 语法, 如 ul 节点的建立, 可以使用 `ul>(li>img)*9`, 然后按 Tab 键来扩展代码, 建立 HTML 结构如下:

```

<!DOCTYPE html>
<html>
<! --head 元素包含了页面的一些信息, 比如 title、meta、样式表和 js-->
<head>
    <meta charset="UTF-8">
    <! --title 内容会显示在浏览器顶端或者作为浏览器标签卡的名称-->
    <title>京东 JD.COM 提供微星 GP 正品行货, 全国价格最低, 放心又轻松</title>

```

```
<! --Favicon:页面图标-->
<link rel="icon" href="images/favicon.ico" type="image/x-icon">
<! --链接外部样式表 -->
<link rel="stylesheet" href="css/template.css">
<! --链接外部 JavaScript 文件-->
<script src="js/template.js"></script>
</head>
<! --body 元素中的所有内容都会在浏览器窗口显示-->
<body>
<! --页面内容容器-->
<div class="show">
  <! --左侧主内容区-->
  <div class="show-left">
    <div class="left-title">
      <div class="title-button"><a href="#">游戏本</a></div>
      <a href="#" class="return">返回商品页</a>
      <h1><a href="#">微星(MSI)GP62MVR</a></h1>
    </div>
    <div class="pic-big">
      <ul id="img-big">
        <li></li>
        <li></li>
        <li></li>
        <li></li>
        <li></li>
        <li></li>
        <li></li>
        <li></li>
        <li></li>
      </ul>
    </div>
  </div>
</div>
<! --左边主内容区结束 -->
<! --右侧缩略图区开始-->
<div class="show-right">
  <div class="list-img">
    <div class="img-title">
      <h2>全部图片</h2>
      <span>(9 张)</span>
    </div>
  </div>
</div>
```



```

<ul class="list-img-show" id="img-small" onclick="change()">
  <li></li>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
</ul>
<div class="turn-page">
  <span class="prev"><b></b></span>
  <span class="page">1/1 页</span>
  <span class="next"><b></b></span>
</div>
</div>
<div class="show-infom">
  <span>京 &nbsp;&nbsp;&nbsp;东 &nbsp;&nbsp;&nbsp;价 &nbsp;&nbsp;&nbsp;:</span>
  <strong class="price">¥ 8099.00</strong>
  <br>
  <span>京东评分:</span>
  <span class="star"></span>
  <div class="clear"></div>
  <div class="buy-btn"><a href="#">加入购物车<b></b></a></div>
</div>
</div>
<! --右侧缩略图区结束-->
<span class="clear"></span>
<! --页面内容容器结束-->
</div>
</body>
</html>

```

2.4.3 编写 CSS 样式

在 WebStorm 项目 css 文件夹中建立样式表文件,命名为“template.css”,双击打开该文件编写样式,代码如下:

```
h1, h2, ul, li, audio {
    margin: 0;
    padding: 0;
    border: 0;
    font: inherit;
}
body {
    margin: 0;
    padding: 0;
    font: 12px/150% Arial, Verdana, "\5b8b\4f53";
}
a {
    text-decoration: none;
}
ol, ul {
    list-style: none;
}
/* 样式初始化部分结束 */
.show {
    width: 990px;
    margin: 10px auto;
}
.show-left {
    width: 770px;
    float: left;
}
.left-title {
    overflow: hidden;
    height: 35px;
    padding: 0 8px;
    border: 1px solid #ddd;
    background: #f7f7f7;
    line-height: 35px;
}
.title-button {
    float: left;
    height: 23px;
    line-height: 23px;
    padding: 0 10px;
    background: #e4393c;
    margin-top: 6px;
```

```
margin-right: 10px;
text-align: center;
color: # fff;
font-weight: bold;
}
.title-button a {
color: # fff;
}
.title-button a:hover {
text-decoration: underline;
}
.return {
color: # 005ea7;
float: right;
width: 69px;
text-align: right;
}
.left-title h1 {
font: 14px '\5fae\8f6f\96c5\9ed1';
height: 35px;
line-height: 35px;
}
.left-title h1 a {
color: # 666;
}
.pic-big {
position: relative;
width: 770px;
height: 800px;
overflow: hidden;
text-align: center;
}
.pic-big img {
max-width: 770px;
}
/* 图片展示左边部分代码完毕 */
.show-right {
width: 210px;
float: right;
}
.list-img {
```

```
border: 1px solid #d0e4c2;
overflow: hidden;
padding: 10px 0 20px;
}
.img-title {
height: 30px;
overflow: hidden;
background-color: #fff;
}
.img-title h2 {
float: left;
padding: 6px 0 0 8px;
font-family: "microsoft yahei";
font-size: 14px;
}
.img-title span {
float: left;
padding-top: 6px;
}
.list-img-show {
overflow: hidden;
padding: 0 0 0 10px;
}
.list-img-show li {
position: relative;
float: left;
width: 90px;
margin: 0 9px 10px 0;
text-align: center;
}
.list-img-show img {
margin: 1px;
padding: 16px;
border: 1px solid #ddd;
cursor: pointer;
}
.list-img-show .hover {
margin: 0;
border: 2px solid #ddd;
}
```

```
.list-img-show img:hover {
    margin: 0;
    border: 2px solid #ddd;
}

.turn-page {
    padding-left: 40px;
}

.turn-page span {
    position: relative;
    float: left;
    height: 14px;
    border: 1px solid #ccc;
    margin-left: 2px;
    border-radius: 5px;
    padding: 3px 11px;
}

.turn-page b {
    position: absolute;
    display: block;
    width: 5px;
    height: 9px;
    background: url("../images/bg_hotsale.gif") no-repeat;
    overflow: hidden;
}

.prev b {
    top: 5px;
    left: 8px;
    background-position: -80px -608px;
}

.turn-page .page {
    width: 60px;
    padding: 4px 0;
    text-align: center;
    border: 0;
    height: 14px;
    line-height: 14px;
    font-size: 12px;
}

.next b {
    top: 5px;
```

```
        right: 8px;
        background-position: -74px -608px;
    }
    .show-infomat {
        overflow: hidden;
        padding: 10px;
    }
    .show-infomat span {
        color: #666666;
    }
    .show-infomat .price {
        color: #E4393C;
        font-weight: bold;
    }
    .show-infomat .star {
        display: inline-block;
        width: 75px;
        height: 14px;
        background: url("../images/newicon20130422.png") -109px -239px no-repeat;
    }
    .buy-btn {
        position: relative;
        padding-top: 10px;
    }
    .buy-btn a {
        position: relative;
        display: block;
        width: 137px;
        height: 38px;
        margin: 0 auto;
        text-align: center;
    }
    .buy-btn b {
        display: block;
        width: 100%;
        height: 100%;
        position: absolute;
        top: 0;
        left: 0;
        background: url("../images/p-btms-new.png") 0 0 no-repeat;
    }
    /* 图片展示右边部分代码完毕 */
```

2.4.4 编写 JS 脚本

在 WebStorm 项目 js 文件夹中建立脚本文件,命名为“template.js”,双击打开该文件编写代码如下:

```
//onload 加载事件;在页面或图像加载完成后执行函数
window.onload = function () {
    //获取鼠标点击的缩略图和对应大图元素
    var tit = document.getElementById("img-small").getElementsByTagName("img");
    var con = document.getElementById("img-big").getElementsByTagName("li");
    //定义鼠标事件 change()事件
    (function change() {
        //通过 for 循环遍历 tit 下所有 img 并绑定单击事件
        for (var i = 0; i < tit.length; i++) {
            tit[i].onclick = imgChange;//定义鼠标事件
        }
    })();
    //定义 imgChange 事件
    function imgChange() {
        for (var i = 0; i < tit.length; i++) {
            //if 判断语句,this 表示当前事件
            if (tit[i] === this) {
                //tit[i]表示鼠标点击时的那个元素,.className 使得这个元素套上 hover 样式
                tit[i].className = "hover";
                //.style.display="block";使得 con[i]这个元素的 display 样式为 block
                con[i].style.display = "block";
            } else { //否则:即为鼠标未点击到的元素
                //tit[i]这个元素的样式为空
                tit[i].className = "";
                //con[i]这个元素的 display 样式为"none"
                con[i].style.display = "none";
            }
        }
    }
};
```

2.4.5 浏览器测试

打开 template.html 文件,将鼠标移至代码视图右上角位置,即会浮出浏览器图标,单击 chrome 浏览器图标,如图 2-6 测试页面所示。

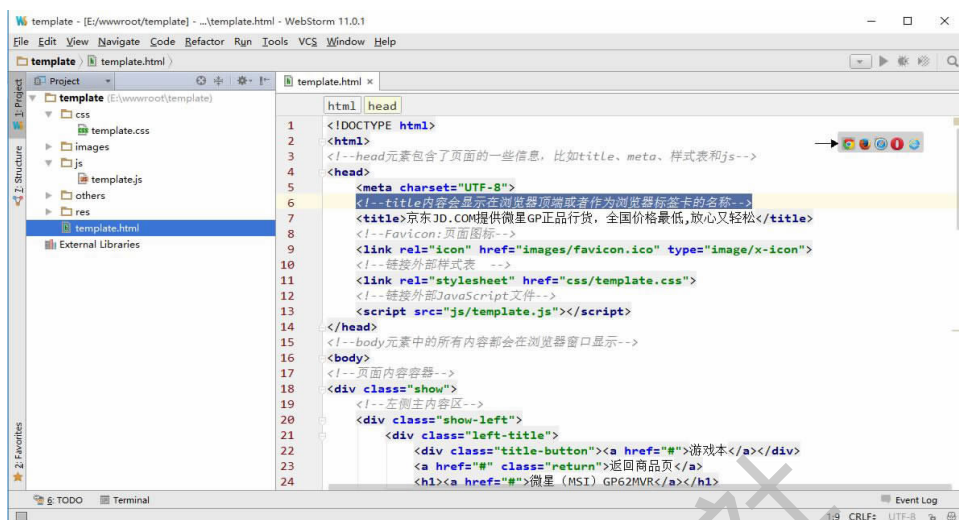


图 2-6 测试页面

Chrome 浏览器启动，显示了 html 文档的内容，如图 2-7 在 Chrome 浏览器测试页面。单击右侧缩略图时，会在左侧显示对应的大尺寸产品图像。

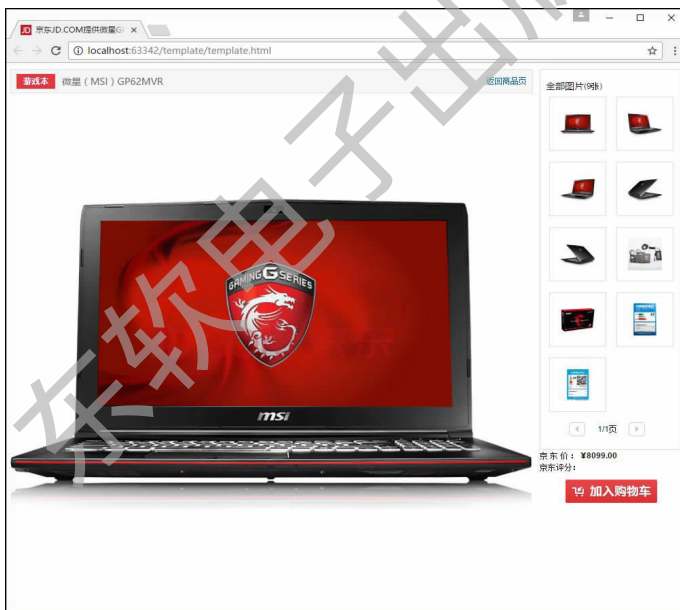


图 2-7 在 Chrome 浏览器测试页面

2.5 强化训练

参考本任务实施过程，从京东官网挑选自己喜欢的商品详情页面，分析 html 结构、CSS 样式和 JS 代码，按建立项目、编写 HTML 文件、编写 CSS 样式、编写 JS 脚本和浏览器测试来实施。