### 拓展知识

### 一、使用注释

在JavaScript中使用注释可用于提高代码的可读性，JavaScript 不会执行注释。按注释的规范，包含有文件注释、普通注释和文档注释。

#### 1．文件注释

文件注释位于文件的最前面，应包括文件的以下信息：

* 概要说明及版本（必须）
* 项目地址（开源组件必须）
* 版权声明（必须）
* 开源协议（开源组件必须）
* 版本号（必须）
* 修改时间（必须）

例如：

/\*!

\* jRaiser 2 Javascript Library

\* sizzle - v1.9.1 (2013-03-15T10:07:24+0800)

\* http://jraiser.org/ | Released under MIT license

\*

\* Include sizzle (http://sizzlejs.com/)

\*/

如果文件内包含了一些开源组件，则必须在文件注释中进行说明。

#### 2．普通注释

普通注释是为了帮助开发者和阅读者更好地理解程序，不会出现在API文档中。其中，单行注释以“//”开头；多行注释以“/\*”开头，以“\*/”结束。

普通注释的使用需遵循以下规定。

* 总是在单行注释符后留一个空格。

例如：// this is comment

* 总是在多行注释的结束符前留一个空格（使星号对齐）。
* 不要把注释写在多行注释的开始符、结束符所在行。

例如：

/\*

这是第一行

这是第二行

\*/

* 不要编写无意义的注释。
* 如果某段代码有功能未实现，或者有待完善，必须添加“TODO”标记，“TODO”前后应留一个空格。

例如：

// TODO 未处理IE6-8的兼容性

function setOpacity(node, val) {

node.style.opacity = val;

}

#### 3．文档注释

文档注释将会以预定格式出现在API文档中。它以“/\*\*”开头，以“\*/”结束，其间的每一行均以“\*”开头（均与开始符的第一个“\*”对齐），且注释内容与“\*”间留一个空格。

例如：

/\*\*

\* 把源对象的属性扩展到目标对象

\* @method extend

\* @param {Any} target 目标对象

\* @param {Any\*} [source] 源对象。若有同名属性，则后者覆盖前者

\* @return {Any} 目标对象

\*/

### 二、常用的正则表达式

正则表达式的结构复杂，但使用范围又较为广泛，为了便于查询和使用，下面总结了一些常用的正则表达式。如表3-10所示。

表3-10 常用正则表达式

|  |  |
| --- | --- |
| **表达式** | **描述** |
| [u4e00-u9fa5] | 匹配中文字符 |
| [^x00-xff] | 匹配双字节字符(包括汉字在内) |
| n[s| ]\*r | 匹配空白行，可以用来删除空白行 |
| w+([-+.]w+)\*@w+([-.]w+)\*.w+([-.]w+)\* | 匹配Email地址 |
| [a-zA-z]+://[^s]\* | 匹配网址URL |
| d{3}-d{8}|d{4}-d{7} | 匹配国内电话号码 |
| [1-9][0-9]{4,} | 匹配腾讯QQ号，腾讯QQ号从10000开始 |
| [1-9]d{5}(?!d) | 匹配中国邮政编码 |
| d+.d+.d+.d+ | 匹配ip地址 |
| ^[1-9]d\*$ | 匹配正整数 |
| ^-?[1-9]d\*$ | 匹配整数 |
| ^[A-Za-z0-9]+$ | 匹配由数字和26个英文字母组成的字符串 |
| ^[a-zA-Z]w{5,17}$ | 匹配以字母开头，长度在6-18之间，  只能包含字符、数字和下划线 |