

梳理正则常用知识点



jCodeLife

2020.08.21 10:55:28 字数 1,049 阅读 18

编辑文章

正则表达式是用于匹配特殊字符和特殊搭配的字符串的强大工具

正则的两种创建方式

1. 字面量

格式: `/pattern/attributes`

```
1 | var reg = /abc/g //表示全局匹配abc字符串
```

2. 通过new RegExp创建

格式: `new RegExp(pattern,attributes)`

```
1 | var reg = new RegExp('abc','g')
```

说明:

pattern表示正则表达式的模式,即要匹配的规则

attributes是一个可选参数,包括 **g i m**, 分别表示全局匹配,忽略大小写和多行匹配。默认情况是非全局匹配、不忽略大小写和单行匹配

返回值是一个具有指定模式也标志的**RegExp**对象, `typeof reg` 为**object**

当**pattern**为不合法正则或者**attributes**包含**gim**以外的字符时,会报错**syntaxError**

下面具体来看一些规则

1. `[]` 用于查找某个范围内的一个字符

例如:

`[abc]` 表示查找abc中的任意一个字符

`[^abc]` 表示查找不是a不是b也不是c的任意一个字符; `^`在`[]`里表示非

`[0-9]` 表示查找从0到9的任意数字, `-`表示到或至的意思

`[a-z]` 表示查找a到z的字符

`[A-Z]` 表示查找A到Z的字符

`[A-z]` 表示查找A到z的字符

2. 元字符 表示拥有特殊含义的一个字符

例如:

`.` 表示任意一个字符,除换行;可用 `\.` 表示`"."`这一个字符

`\w` 表示单词字符

`\W` 表示非单词字符

`\d` 表示数字字符

`\D` 表示非数字字符

`\s` 表示查找空白字符

`\S` 表示查找非空白字符

`\n` 换行

`\r` 回车符

`\t` 制表符

推荐阅读

Es6基础语法

阅读 141

为什么总有人觉得前端很简单?

阅读 120

vue第2天 (vue生命周期钩子、日期格式化库 moment.js、vue计算属...

阅读 51

面试了几个前端,给爷整哭了!

阅读 50,309

浅谈JS原型链

阅读 60

也可在后续处理中单独获得这一部分正则的匹配结果。小括号将其作为一个组，同时记下了匹配的结果。正则表达式 `\n` 表示“第N个组”的结果，因为一个正则表达式中可能有多个小括号扩起的组，所以用 `\1 \2 \3` 等来分别代表之前的第1组第2组第3组。

4. 量词 表示字符出现的数量

+ 表示一个到多个

* 表示0个到多个

? 表示存0个或1个

{x} 表示出现x个的字符串

{x,y} 表示x个到y个的字符串

{x,} 表示x个到infinity个的字符串

\$ 表示以什么结尾的字符串，比如/`n$`/表示以n结尾的字符串

^ 表示以什么开头的字符串，比如/`^n`/表示以字母n开头的字符串

?= 表示后面紧接某字符串的字符串，比如/`?=n`/表示后面紧接字母n的字符串

?! 表示后面不紧接某字符串的字符串，比如/`?!n`/匹配后面不紧接n的字符串

| 表示或者，例如(`x|y`) 匹配x或者y

5. 正则对象常用方法

- `compile()` 编译和改变正则
- `test()` 用于检测某字符串是否包含正则表达式所表示的字符串。返回true或false
- `exec()` 检测某字符串是否包含正则表达式所表示的字符串，返回一个对象，这个对象记录了找到的值和第一次出现的位置 `index` 下标

```
> /e/.exec("The best things in life are free!");
< ▼ ["e", index: 2, input: "The best things in life are free!", groups: undefined] ⓘ
  0: "e"
  groups: undefined
  index: 2
  input: "The best things in life are free!"
  length: 1
  ▶ __proto__: Array(0)
```

6. 包装类字符串对象身上的常用方法(与正则相关的)

- 在`replace()` 中使用正则
表示替换字符串中通过正则匹配的子串

```
1 var str = "Visit Microsoft!";
2 var res = str.replace(/microsoft/i, "W3School");
3 //res 的结果将是: Visit W3School!
```

- `search` 查找字符串中与正则相匹配的子字符串在字符串中的索引位置，没有匹配到返回-1

```
> var str = "afafjaskdfhakshavcab123abc"
< undefined
> str.search(/abc/g);
< 18
> str.search(/abcd/g);
```

- `match` 用于返回与正则相匹配的子串，返回子串组成的类数组

```
▶ var str = "afafjaskdfhakshavcab123abc"
▶ undefined
▶ str.match(/af/g)
▶ ▶ (2) ["af", "af"]
▶ typeof str.match(/af/g)
```

推荐阅读

Es6基础语法

阅读 141

为什么总有人觉得前端很简单?

阅读 120

vue第2天 (vue生命周期钩子、日期格式化库 moment.js、vue计算属...

阅读 51

面试了几个前端，给爷整哭了!

阅读 50,309

浅谈JS原型链

阅读 60

"小礼物走一走，来简书关注我"

赞赏支持

还没有人赞赏，支持一下



jCodeLife  书山有路勤为径，学海无涯苦作舟
总资产17 (约1.50元) 共写了12.9W字 获得251个赞 共23个粉丝



写下你的评论...

全部评论 0 只看作者 关闭评论

按时间倒序 按时间正序

被以下专题收入，发现更多相似内容

 投稿管理

+ 收入我的专题



领跑计划笔记

推荐阅读

Es6基础语法

阅读 141

为什么总有人觉得前端很简单？

阅读 120

vue第2天（vue生命周期钩子、日期格式化库 moment.js、vue计算属...

阅读 51

面试了几个前端，给爷整哭了！

阅读 50,309

浅谈JS原型链

阅读 60



写下你的评论...



评论0



赞1

