

正则表达式的模式匹配

课程介绍

说文解字--正则表达式的模式匹配

- 1、模式：规则
- 2、匹配：作比较
- 3、正则表达式都是操作字符串的

课程介绍

课程内容

- 1、第一次亲密接触
- 2、第二次深入了解
- 3、常用的正则表达式

请大家认真完成课后练习

正则表达式的第一次亲密接触

什么是正则表达式

由以下两种字符组成：

- 1.普通字符（例如26个英文字母、数字等）
- 2.特殊字符（有特殊含义的，例如.\等）。

说明：

该模式描述在查找文字主体时待匹配的一个或多个字符串。

正则表达式作为一个模板，将某个字符模式与所搜索的字符串进行匹配

。

正则表达式的第一次亲密接触

为什么使用正则表达式

1、查找

2、替换

3、数据有效性验证

课程总结

正则表达式能做什么

- 1、查找特定字符
- 2、替换文本(eg.敏感词过滤)
- 3、数据有效性验证

课程总结

创建正则表达式的两种方式

1、字面量或直接量：`/js/igm`

2、构造函数：`new RegExp('js', 'igm')`

`new RegExp(变量, 'igm')`

课程总结

模式修饰符—g 全局匹配

exec: 无论是否全局匹配，都会返回分组中匹配到的内容；

无论是否全局匹配都只返回当前匹配到的内容，而不是所有，

只不过全局匹配时可以继续匹配下去。

课程总结

模式修饰符—g 全局匹配

match: 只有非全局匹配，才会返回分组中匹配到的内容；

全局匹配只会一次返回所有匹配到的字符。

课程总结

模式修饰符—g 全局匹配

replace : 非全局匹配 , 只替换第一个匹配到的内容 ; 全

局匹配替换所有匹配到的内容

test、search、split : 没区别

课程总结

模式修饰符—m 多行匹配

1.js

2.js

3.js

1.全局匹配

2.使用^或\$

课程总结

简单的转义字符

- 1、`\n` 换行符
- 2、`\t` 制表符
- 3、`\xnn` 由十六进制数nn指定的拉丁字符(eg. `\x0A` \leq `\n`)
- 4、`\uxxxx` 由十六进制数xxxx指定的Unicode字符(eg. `\u0009` \leq `\t`)

课程总结

字符类

- 1、`[...]`和`[^...]` 方括号内的任意一个字符
- 2、`.` 除换行符之外的任意一个字符
- 3、`\w`和`\W` 任何字母、数字以及下划线组成的一个字符
- 4、`\d`和`\D` 任何一个数字
- 5、`\s`和`\S` 任何Unicode空白符(eg.空格、制表符、`\n`)

课程总结

重复(量词)

1、 $\{n,m\}$ $n \leq x \leq m$

2、 $\{n, \}$ $x \geq n$

3、 $\{n\}$ $x = n$

4、 $? \{0,1\}$

5、 $+ \{1, \}$

6、 $* \{0, \}$

课程总结

非贪婪匹配

1、 $\{n,m\}?$

2、 $\{n,\}$?

3、 $\{n\}$?

4、 $??$

5、 $+$?

6、 $*$?

课程总结

选择、分组和引用

- 1、| 选择，匹配该符号左边或右边的东西
- 2、(...) 分组 统一操作；捕获括号中的内容；
- 3、(?:...) 非捕获性分组
- 4、\n 第n个捕获性分组的引用 (eg.\1 \2 \3...)

课程总结

捕获到的内容

- 1、exec返回的数组中
- 2、/1/ 模式中
- 3、replace的第二个参数中，\$1
- 4、RegExp.\$1

课程总结

指定匹配位置

1、`^` 首匹配

2、`$` 尾匹配

3、`\b`和`\B` 单词边界，即`\w`和`\W`之间的位置

4、`(?=p)`和`(?!p)` 前瞻性匹配，只有接下来的字符`=p`，才匹配之前的内容

课程总结

RegExp对象的实例方法

1、test(字符串)

2、exec(字符串)

3、toString()、toLocaleString()、valueOf()

课程总结

RegExp对象的实例属性

- 1、`global` 是否设置了g
- 2、`ignoreCase` 是否设置了i
- 3、`multiline` 是否设置了m
- 4、`source` 字面量形式对应的字符串
- 5、`lastIndex` 开始搜索下一个匹配项的字符位置，默认0

课程总结

RegExp对象的构造函数属性

- 1、input 匹配的字符串
- 2、lastMatch 最近一次的匹配项
- 3、lastParen 最近一次匹配的捕获组
- 4、leftContext 字符串中匹配到的内容之前的文本
- 5、rightContext 字符串中匹配到的内容之后的文本
- 6、\$1~\$9 第1~第9个匹配到的捕获组

课程总结

String对象和正则表达式相关的方法

- 1、search(正则或字符串)
- 2、match(正则)
- 3、split(正则或字符串)
- 4、replace(正则或字符串, 替换成的文本或函数)

谢谢！