

正则

正则方法

正则符号

正则例题

# 正则

- 定义：专门用来检索模糊范围的一种规则
- 书写方法：
  - //
  - 实例化：new RegExp('字符串', '修饰符');
- 正则特征：
  - **懒惰性**: \d 让它找一个它一定不会找第二个
  - **贪婪性**: \d+ 有多少就尽量匹配多少

## 正则方法

- exec
  - /规则/.exec(字符串)
  - 找到正则中第一个匹配的字符且放到数组中。
  - 返回值：数组
  - [字符、index、input、groups]
- test
  - /规则/.test(字符串)
  - 检测正则是否匹配字符串
  - 返回值：布尔值 成功为true，否则为false
- str.match(/正则/)
  - 找到正则中所有匹配的字符且放到数组中。
  - 返回值：数组 没有返回null
- string.replace
  - string.replace(要替换的字符串||正则,替换成什么||函数);
  - 当第二个参数为函数的时候
    - 函数的第一个参数就是每次匹配到的结果
    - 函数的第二个参数index
    - 函数的第三个参数input
    - 函数的第四个（及以后）参数undefined
  - 返回值：新的字符串

- 注：此函数必须要有return，不然默认为undefined

# 正则符号

- \ 转义符
  - 注意：\*需要注意的是在写 / 记得转义
- 元字符 (+字母有特殊的含义)
  - \n 换行
  - \d 一个数字
  - \D 一个非数字
  - \s 一个空格
  - \S 一个非空格
  - \w 一个数字字母下划线
  - \W 一个非数字字母下划线
  - \b 一个边界符
  - \B 一个非边界符
  - \r\n 回车并换行
  - . 任意一个字符，除回车
  - | 或者
- 修饰符
  - g 全局
  - i 忽略大小写
  - m 换行匹配、多行匹配（加开头或结尾）
- 量词
  - + 最少一个，最多不限
- 小括号 () —— 子项
  - 作用1: () 提权  $(1+1)^*1$
  - 作用2: 子项能够在一个规则中提取某些(指定)字符
  - 子项的顺序是从左往右数的，每有一个()就是一个子项
  - 从第二个参数起子项与形参一一对应。
  - 注意：子项如果包了一个规则，在规则之后有量词，那么结果为最后一个
    - 比如:let str = '2019'
    - $(\d)^+$  -> 找到的数字不是2是9
- 子项重复项
  - \+ 数字
  - 例:  $/(d1)c1(b1)\1\2/$   
当前有2个子项，一个是d1另一个是b1  
 $\1$ 就代表第一个子项， $\2$ 就代表第二个子项
- []
  - 在中括号中任意选择一个字符
  - 比如: [12345] 可以为1也可以为2也可以为3...也可以简写为[1-5],因为 - 是1到5的

意思

- 表示方法：
  - [0-9] 数字
  - [A-Z] 大写英文字母A-Z
  - [a-z]小写英文字母a-z
  - [\u4e00-\u9fa5]中文区间范围
- 开始结束符
  - ^开始符号
  - \$结束符号
- [^] 排除符号
  - 比如: [^36543] 排除 36543
- {} 范围
  - {n,m} 最小n次, 最大m次
  - {n} 最小n次, 最大n次
  - + {1, }最小1次, 最多无限
  - ? {0,1} 最小0次, 最多1次
  - \*号 {0, } 最小0次, 最多无限次
- QQ表示方法:
  - 首字母不能是0, 全部都要是数字, 最小5位,最多11位  
`/^[1-9]\d{4,10}$/`
- 手机号:
  - `1[3456789]\d{9}`
- 身份证号:
  - `/^省市區((1[89][0-9][0-9])|(20[0-1][0-9]))(0[1-9]|1[0-2])(0[1-9]|[1-2][0-9]|3[01])\d{3}[\dx]$/`

## 正则例题

- 把下列字符串为数字的找出来（包括连续的），并且存到一个数组中？

```
let str = 'y28nu82u3l8rixmir4d83f2x';
console.log(str.match(/\d+/g));
```

- 检测当前str是否属于18-108之间？

```
console.log(/^(1[89]|[2-9][0-9]|10[0-8])$/ .test(str));
```

- 计算下面字符串中的字节长度，假设一个英文字节为1，一个中文为2

```
let str = 'hello!~张欢,您老贵庚?'; //21
let num = 0;
for(let i=0;i<str.length;i++){
    if(/[\u4e00-\u9fa5]/.test(str[i])){
        num += 2;
    }else{
        num ++
    }
}
console.log(num);
```

- 下划线的使用

```
let str2 = 'a231_';
console.log(/^[A-Za-z]\w{5,17}$/);
```