

task21

问答

1. `\d`, `\w`, `\s`, `[a-zA-Z0-9]`, `\b`, `.`, `*`, `+`, `?`, `x{3}`, `^`, `$` 分别是什么?

`\d`: 查找数字

`\w`: 查找单词字符 (字母, 数字, 下划线)

`\s`: 查找空白字符 (空格, 换行, tab, 回车)

`[a-zA-Z0-9]`: 查找所有大小写字母, 数字0-9

`\b`: 匹配单词边界

`.`: 匹配除了换行和回车的任意字符

`n*`: 匹配任何包含零个或多个n的字符串

`n+`: 匹配任何至少包含一个n的字符串

`n?`: 匹配任何包含零个或一个n的字符串

`x{3}`: 匹配包含3个x的序列的字符串

`^n`: 匹配任何开头为n的字符串

`n$`: 匹配任何结尾为n的字符串

2. 贪婪模式和非贪婪模式指什么?

贪婪与非贪婪模式影响的是被量词修饰的子表达式的匹配行为, 贪婪模式在整个表达式匹配成功的前提下, 尽可能多的匹配, 而非贪婪模式在整个表达式匹配成功的前提下, 尽可能少的匹配。非贪婪模式只被部分NFA引擎所支持。

属于贪婪模式的量词, 也叫做匹配优先量词, 包括:

`{m,n}`、`{m,}`、`?`、`*` 和 `+`。

在一些使用NFA引擎的语言中, 在匹配优先量词后加上 `?`, 即变成属于非贪婪模式的量词, 也叫做忽略优先量词, 包括:

`{m,n}?`、`{m,}?`、`??`、`*?` 和 `+`。

代码题

1. 写一个函数 `trim(str)`, 去除字符串两边的空白字符

```
function trim(str) {  
    return str.replace(/^\s+|\s+$/g, "");  
}
```

2. 使用实现

`addClass(el, cls)`、`hasClass(el, cls)`、`removeClass(el, cls)`

使用正则

ps：视频里问题纠正

```
/*
\b指的是单词边界，包括 开头、结尾、空格分割、中横线分割
所以对于 hunger-valley 中的 hunger 会被认为是一个单词，但hunger-valley整体才是一个
class，所以 hasClass的判断不能用 \b
*/
function hasClass(el, cls) {
    var reg = new RegExp('(\\s|^)'+cls+'(\\s|$)')
    return reg.test(el.className);
}

//提示：el为dom元素，cls为操作的class，el.className获取el元素的class
```

3.写一个函数 `isEmail(str)`，判断用户输入的是不是邮箱

```
var str='code.zhangrui@gmail.com';
function isEmail(str){
    var reg = /^[\\S]+[@]{1}[a-zA-Z0-9]{1,}\\.(com|cn)/g;
    console.log(reg.test(str));
}
isEmail(str);
```

4.写一个函数 `isPhoneNum(str)`，判断用户输入的是不是手机号

```
var str='18878988888';

function isPhoneNum(str){
    var reg = /1\d{10}/g;
    console.log(reg.test(str));
}

isPhoneNum(str);
```

5.写一个函数 `isValidUsername(str)`，判断用户输入的是不是合法的用户名（长度6-20个字符，只能包括字母、数字、下划线）

```
var str='r_89fu4jff8y4943939n';

function isValidUsername(str){
    var reg = /^[\\w]{6,20}$/g;
```

```
console.log(reg.test(str));  
}  
  
isValidUsername(str);
```

6.写一个函数 `isValidPassword(str)`，判断用户输入的是不是合法密码（长度6-20个字符，只包括大写字母、小写字母、数字、下划线，且至少至少包括两种）

```
var str='r_89fu4jff8y4943939n';  
  
function isValidPassword(str){  
    var reg = /^(^([A-Z]{6,10}$)|(^([a-z]{6,10}$)|(^([0-9]{6,10}$)|(^_{6,10}$)))/;  
    if(!reg.test(str)){  
        return 'nothing wrong';  
    }  
    else{  
        return 'error';  
    }  
}  
  
isValidPassword(str);
```

7.写一个正则表达式，得到如下字符串里所有的颜色（`#121212`）

```
var re = /[[:|\s]\#[A-Z|a-z|0-9]{6}/g  
  
var subj = "color: #121212; background-color: #AA00ef; width: 12px; bad-colors: f#fddee #fd2 "  
  
alert( subj.match(re) )
```

8.下面代码输出什么？为什么？改写代码，让其输出 `hunger, world`.

```
var str = 'hello "hunger" , hello "world";  
var pat = /".*"/g;    //贪婪模式  
str.match(pat);  
  
var str = 'hello "hunger" , hello "world";  
var pat = /".*" ,/g;    //贪婪模式  
str.match(pat);  
//[ ""hunger" ," ]
```

```

var str = 'hello "hunger" , hello "world" ,';
var pat = /\.*/g; //贪婪模式
str.match(pat);
//[[""hunger" , hello "world" ,"]]

var str = 'hello "hunger" , hello "world"';
var pat = /\.?"/g; //非贪婪模式
str.match(pat);
//hunger, world

var str = 'hello "hunger" , hello "world"';
var pat = /"[^"]+"/g; //贪婪模式
str.match(pat);

```

9. 补全如下正则表达式，输出字符串中的注释内容。(可尝试使用贪婪模式和非贪婪模式两种方法)

```

str = '.. <!-- My -- comment \n test --> .. <!--> .. ';
re = /<[^\r]*?>/g; //非贪婪模式1
console.log(str.match(re)); // '<!-- My -- comment \n test -->', '<!-->'

str = '.. <!-- My -- comment \n test --> .. <!--> .. ';
re = /<[\W\w]*?>/g; //非贪婪模式2
console.log(str.match(re));

str = '.. <!-- My -- comment \n test --> .. <!--> .. ';
re = /<[^]*?>/g; //非贪婪模式3
console.log(str.match(re));

str = '.. <!-- My -- comment \n test --> .. <!--> .. ';
re = /<!--[^\>]*-->/g; //贪婪模式1
console.log(str.match(re));

str = '.. <!-- My -- comment \n test --> .. <!--> .. ';
re = /<[^\.]*>/g; //贪婪模式2
console.log(str.match(re)); // '<!-- My -- comment \n test -->', '<!-->'

str = '55 <!-- My -- comment \n test --> 55 <!--> 55 ';
re = /<[^\.]*>/g; //测试
console.log(str.match(re));
//[["55 <!-- My -- comment \n test --> 55 <!-->"]]
//这说明[]中的.只是个普通点，不是文本什么的
//2个贪婪模式的例子就是说，在右边的那个>!-->之前，不能有./>

```

10. 补全如下正则表达式

```

var re = /<\w.*?>/g;

```

```
var str = '<> <a href="/"> <input type="radio" checked> <b>';  
console.log(str.match(re)); // '<a href="/">', '<input type="radio" checked>',  
'<b>'
```



Leanote
Upgrade Account