# 一、昨日回顾



修饰符：

g：加入修饰符g，表示执行全局匹配。不加g，表示只查找一次。

i：表示不区分大小写。

正则对象的方法：

正则.test(字符串);

正则.exec(字符串);

字符串对象的方法：

字符串.match(正则)

字符串.search(正则);

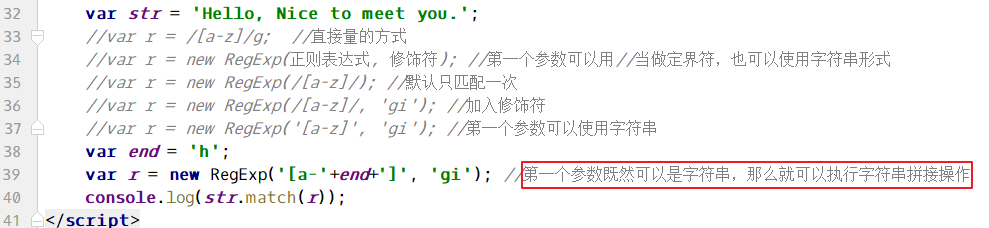
字符串.split(正则);

字符串.replace(正则, 替换成什么); //第二个参数可以是一个值，也可以是一个函数

上述各个方法中，使用的正则可以是直接量的正则表达式，也可以是实例化后的正则对象：

直接量语法：只能写两条斜线的形式。

实例化正则对象：可以写两条斜线的形式，也可以写字符串的形式。



# 二、今日目标

# 三、正则案例

1、匹配邮箱格式是否正确

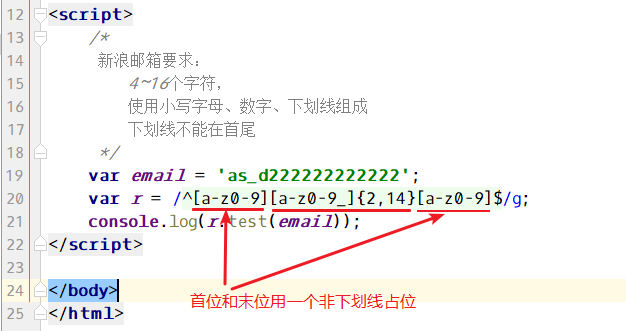
如果说邮箱的@后面的部分如果固定，则只需要验证前面的邮箱名部分。比如新浪。

新浪邮箱要求：

4~16个字符，

使用小写字母、数字、下划线组成

下划线不能在首尾



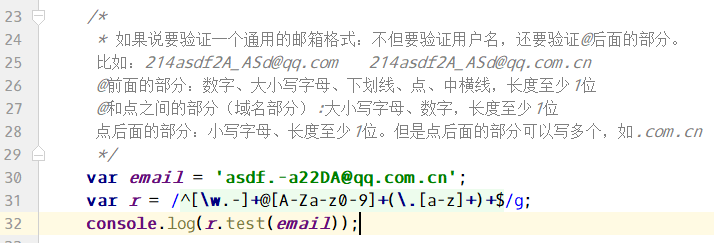
如果说要验证一个通用的邮箱格式：不但要验证用户名，还要验证@后面的部分。

比如：214asdf2A\_ASd@qq.com 214asdf2A\_ASd@qq.com.cn

@前面的部分：数字、大小写字母、下划线、点、中横线，长度至少1位

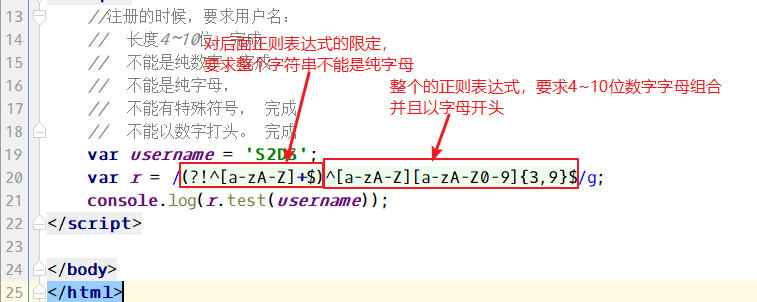
@和点之间的部分（域名部分）:大小写字母、数字，长度至少1位

点后面的部分：小写字母、长度至少1位。但是点后面的部分可以写多个，如.com.cn



2、用户名验证

注册的时候，要求用户名：长度4~10位，不能是纯数字，不能是纯字母，不能有特殊符号，不能以数字打头。



# 四、PHP中的正则表达式

语法和JavaScript中的正则语法基本一致。

PHP中的正则函数：



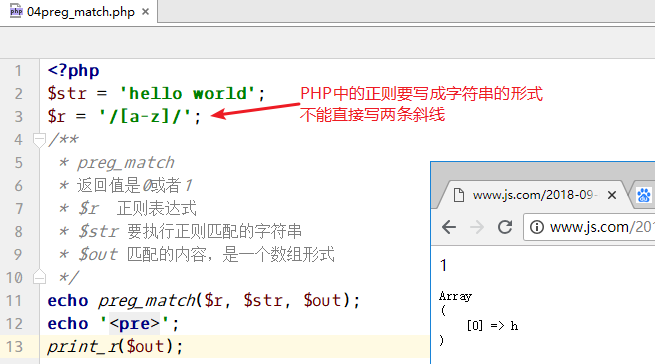
1、preg\_match()

preg\_match()返回匹配的次数，它的返回值是0或者是1.

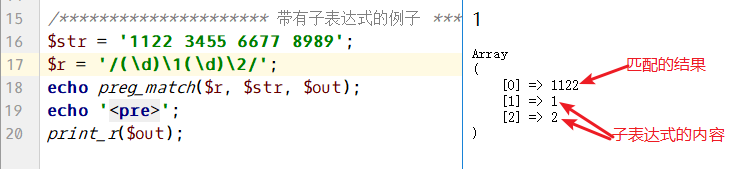
0表示不匹配。

1表示匹配一次，因为preg\_match在匹配一次之后就停止向后检索了。和js中不加g的情况一样。

查看匹配的结果：

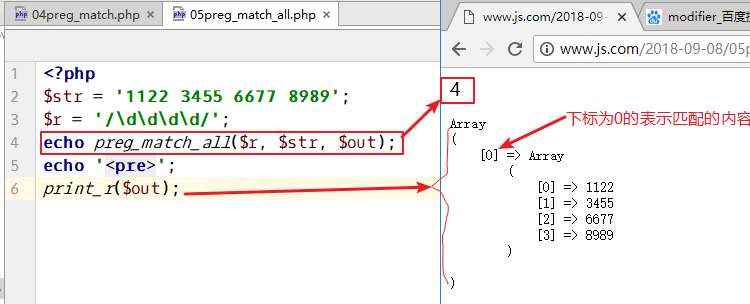


带有子表达式的例子：

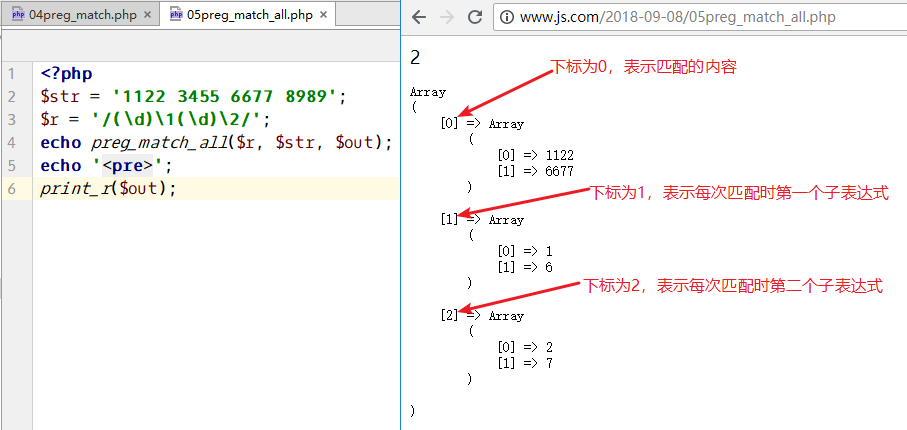


2、preg\_match\_all();

执行正则表达式的全局匹配，类似于js中的带g的匹配。



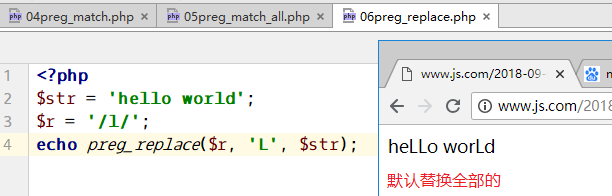
带有子表达式的例子：



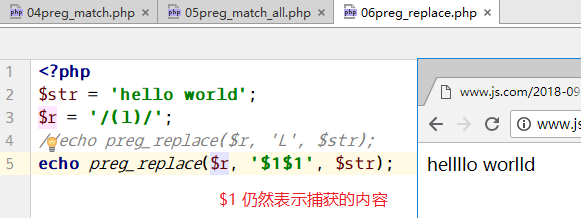
3、preg\_replace();

执行正则替换。

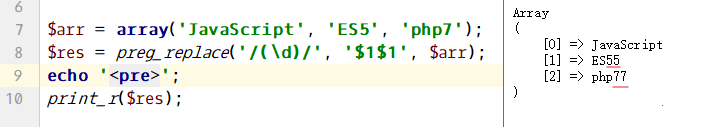
替换字符串：



带有子表达式：

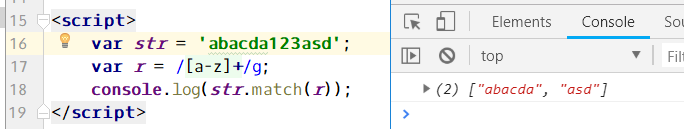


替换数组：



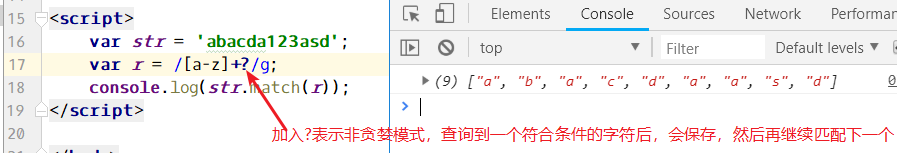
4、案例：解决贪婪问题？

无论是JS中，还是PHP中，正则表达式的匹配模式默认都是贪婪模式。

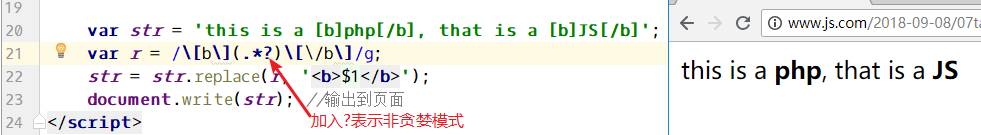


上述代码，查询至少一个字母。但是在实际匹配的过程中，匹配到一个a之后，并没有停止，而是继续向后匹配，得到一个连续的字符串。这就是贪婪模式。

下面在正则表达式后面加入?，表示非贪婪模式：



默认是贪婪模式，必须使用非贪婪模式的例子：



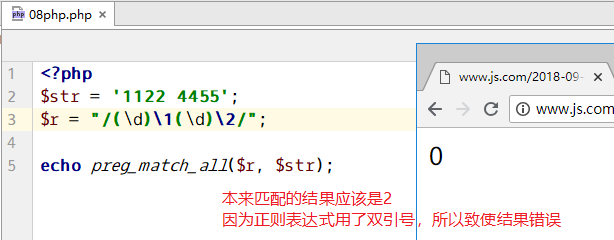
PHP中，强制非贪婪模式：

加入大写的修饰符U即可。



5、PHP中使用正则注意点

PHP中的正则表达式除了//以外，外面还要加引号，这个引号要用单引号。



# 五、正则练习

1、表单提交才进行判断



2、表单提交的时候，判断用户名

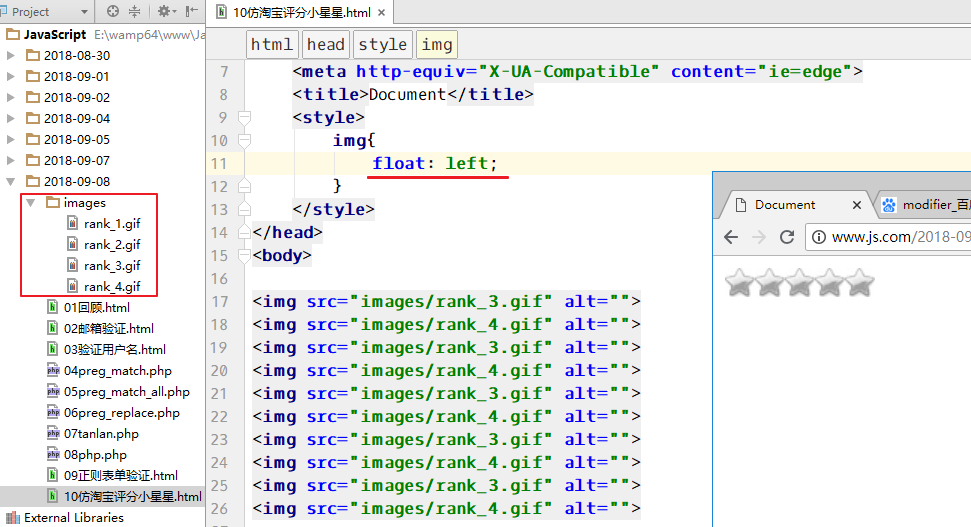


3、判断密码（带密码强度）



# 六、仿淘宝评分小星星

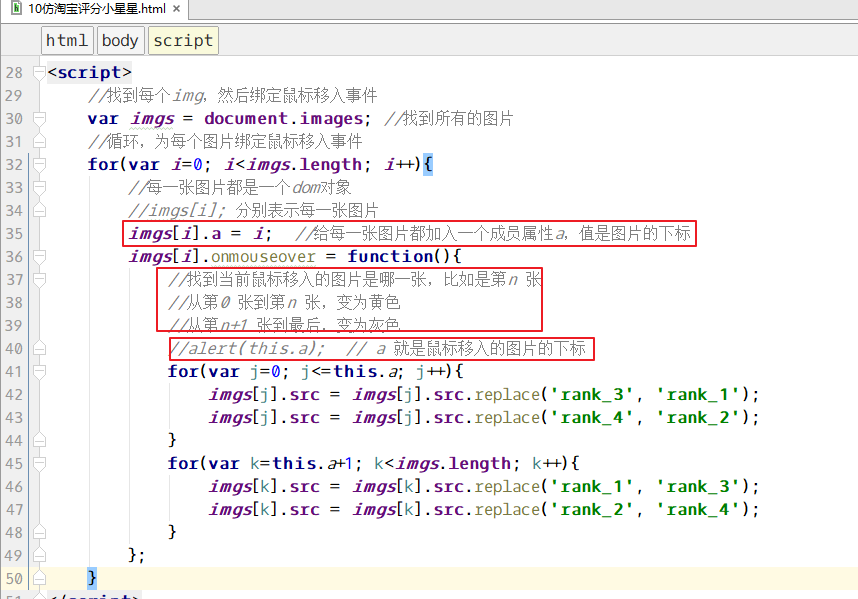
1、HTML+css



2、找到所有图片，绑定鼠标移入事件



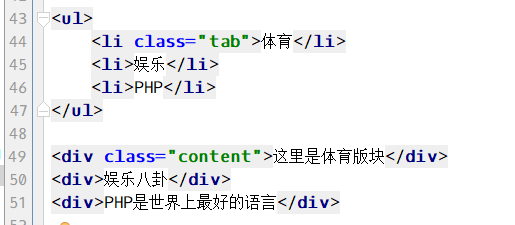
3、找到鼠标移入的图片，设置它左侧和右侧的图片



# 七、标签页效果

1、HTML+css

HTML：



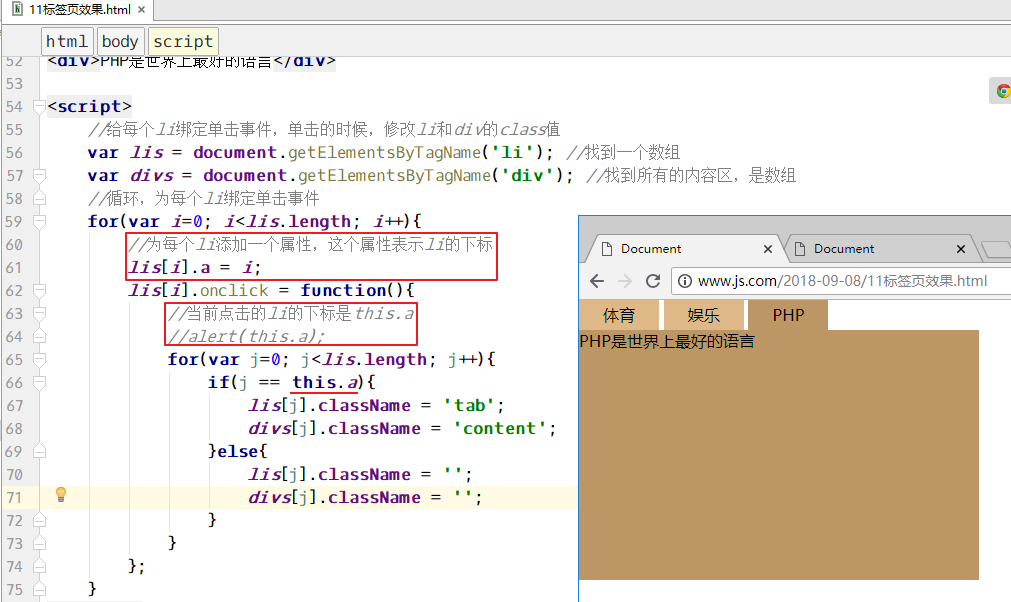
css：



效果：



2、JS代码



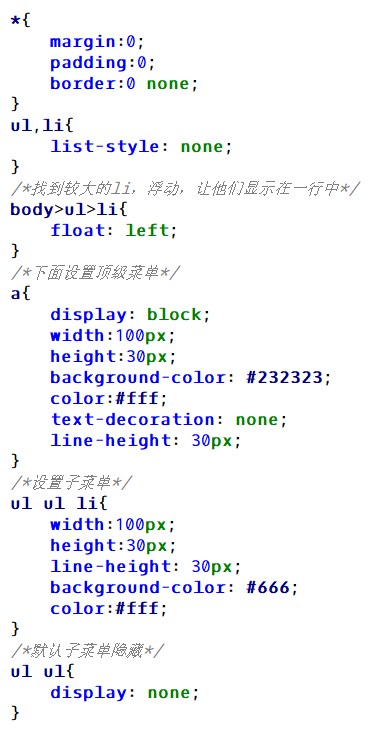
# 八、横向下拉菜单

1、HTML+css

HTML：



css：



2、JS代码



3、让下拉菜单慢慢变化

