正课:

String

1. 获得指定位置的字符

2. 获取子字符串

3. \*\*\*查找关键词

4. \*替换

5. \*切割

\*\*\*\*正则表达式

String

1. 获得指定位置的字符: ——了解

str.charAt(i) => str[i]

php: substr($str, i, 1) 获得$str中i位置的1个字符

获得指定位置字符的unicode号:

str.charCodeAt(i)

将unicode号反向转回字:

String.fromCharCode(unicode号)

2. 获取子字符串:

str.substring(starti,endi+1) => str.slice(starti,endi+1)

substring vs slice

相同: 1. 含头不含尾

2. 省略第二个参数，表示一直选取到结尾

不同: 不支持负数参数！

变通: 用str.length-n 代替 -n

str.substr(starti,n) 根据个数，获取子字符串

php: substr($str, i, n) 获得$str中i位置开始的n个字符

简写: i支持负数参数

省略n表示到结尾

3. 查找关键词: 4种:

1. 查找一个固定的关键词出现的位置:

var i=str.indexOf("关键词",开始位置)

在str中，从“开始位置”开始，查找下一个"关键词"的位置

返回值: 下一个"关键词"的第一个字的下标位置

如果找不到了，返回-1

如果省略第二个参数-开始位置: 默认从0开始找

查找最后一个关键词的位置:

var i=str.lastIndexOf("关键词")

问题: 1. 区分大小写!

2. 只能查找一个固定的关键词!

php: strpos -> indexOf

stripos 忽略大小写版本的indexOf

strrpos -> lastIndexOf

strripos 忽略大小写的lastIndexOf版本

$i=stripos(完整字符串,关键词[,开始位置])

2. \*\*\*\*正则表达式: Regular Expression

什么是: 规定字符串中，字符出现规律的表达式

何时: 2种:

1. 用规则，模糊匹配多种敏感词时

2. 用规则验证字符串的格式: 密码强度，邮箱格式，日期。。。

如何: 语法:

1. 最简单的正则就是关键词原文本身

2. 字符集:

什么是: 1位字符上的备选字符的集合

何时: 只要一位字符上有多种可能的备选字时

如何: [备选字符列表]

强调: 1. 不能加逗号分隔

2. 每个[]，只能匹配一位字符

简写: 如果字符集中部分字符的unicode号是连续的，可用-省略中间字符

常用: [0-9] 一位数字

[a-z] 一位小写字母

[A-Z] 一位大写字母

[A-Za-z] 一位字母，大小写都行

[0-9A-Za-z] 一位字母或数字

[\u4e00-\u9fa5] 一位汉字

车牌号: [\u4e00-\u9fa5] [A-Z] [A-Z0-9] [A-Z0-9] [A-Z0-9] [A-Z0-9] [A-Z0-9]

除了: [^74]

匹配范围太大！

比如匹配HTML元素的开始标签: <a[^>]+>

3. 预定义字符集:

什么是: 对常用字符集的简写: 固定4种:

包括: \d 1位数字

\w 1位字母，数字或\_

\s 1位空字符: 空格，tab，...

. 一切字符

强调: 只有规则和预定义字符集的规则完全匹配时，才能使用预定义字符集

4. 量词:

什么是: 专门规定一位字符集出现的次数

何时: 今后只要固定字符集的出现次数时，必须用量词

如何: 字符集量词

强调: 量词默认只修饰相邻的前一个字符集

包括: 2大类:

1. 有明确数量:

{n,m} 至少n个，最多m个

{n,} 至少n个，多了不限

{n} 必须n个，不能多也不能少

2. 没有明确数量:

? 可有可无，最多1个

\* 可有可无，多了不限

+ 至少1个，多了不限