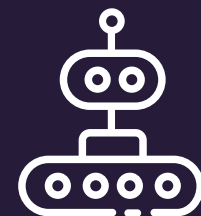


入门篇



正则表达式

@鸽婆打字机



元字符-1

^ 行的起始
\$ 行的结束

`^cat$`

[...] 字符组
[^...] 排除型字符组
[a-z0-9] 区间

`gr[ae]y, [Ss]treet`
`q[^u]`
`H[1-6]`

. 单个任意字符
? * + 0~1, 0~n, 1~n
{, } 重复次数

`2021.2.5`
`.?`
`[0-9]{6,9}`

元字符-2

|

多选结构

(First|1st)

\

转义符

\.txt, \\$

(...)\1

反向引用

(the).+\1

\b

单词分界符

\bcat\b

\B

非单词分界符

元字符-3

<code>\t</code>	制表符
<code>\n</code>	换行符
<code>\r</code>	回车符
<code>\s</code>	空白字符(包括空格符、制表符、换行符、回车符)
<code>\S</code>	非空白字符
<code>\w</code>	单词中的字符,即 <code>[a-zA-Z0-9]</code> (<code>\w+</code> 可用来匹配一个单词)
<code>\W</code>	非单词字符,即 <code>[^a-zA-Z0-9]</code> 或 <code>[^\w]</code>
<code>\d</code>	数字,即 <code>[0-9]</code>
<code>\D</code>	非数字字符,即 <code>[^0-9]</code> 或 <code>[^\d]</code>

类型	元字符	含义	匹配对象
基础	.	单字符	单个任意字符
	\	转义符	三个作用: (1) 将元字符变成对应的普通字符, (2) 将非元字符变成元字符序列, (3) 匹配普通字符 (被忽略)
	()	括号	三个作用: (1) 限制多选结构, (2) 分组, (3) 捕获文本
量词	?	可选项	0~1 个前面紧邻的单元
	*	任意次数	0~n 个前面紧邻的单元
	+	至少一次	1 到 n 个前面紧邻的单元
	{ min, max }	重复区间	min~max 个前面紧邻的单元
结构	[...]	字符组	单个列出的字符
	[^...]	排除型字符组	单个未列出的字符
	[a-z0-9]	区间	注意: 连字符 - 在字符组内是元字符, 字符组外不是元字符
	(...)	子表达式	受量词作用的单元
	(... ...)	多选结构	竖线分隔两边的任意一个子表达式
	(...)(...)\1\2	反向引用	用 \1、\2 等表示第一、第二组括号匹配的文本
分界符	\b...\b	单词分界符	单词的起始和结束
	\<...\>		
	^...\$	行分界符	行的起始和结束



进阶篇？技巧+Python实操

四种类型的环视？

命名捕获？（?P<Name>…）

性能测试与优化？

消除循环？

○○○○○○

Bye bye!!!

资料

① 书籍：

精通正则表达式 (*Mastering Regular Expressions*)

正则表达式必知必会 (*Sams Teach Yourself Regular Expressions in 10 Minutes*)

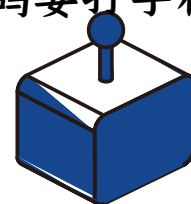
正则表达式经典实例 (*Regular Expressions Cookbook*)

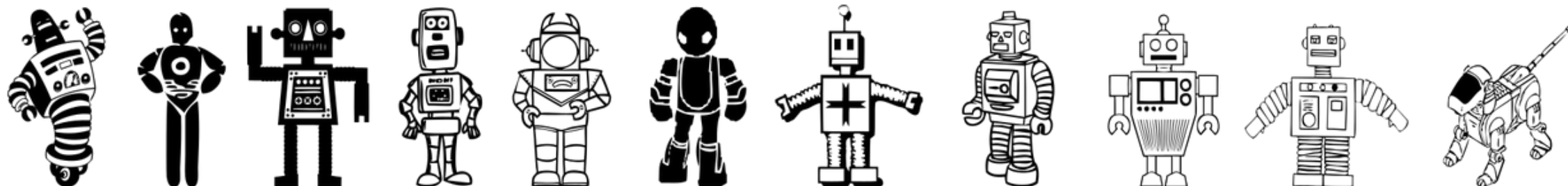
② 网站：

<https://regex101.com/>

<https://regexr.com/>

@鸽婆打字机





鸽婆打字机

