写给前端的正则表达式入门

savokiss SegmentFault 今天

本文转载于 SegmentFault 社区专栏：码力全开

作者：savokiss

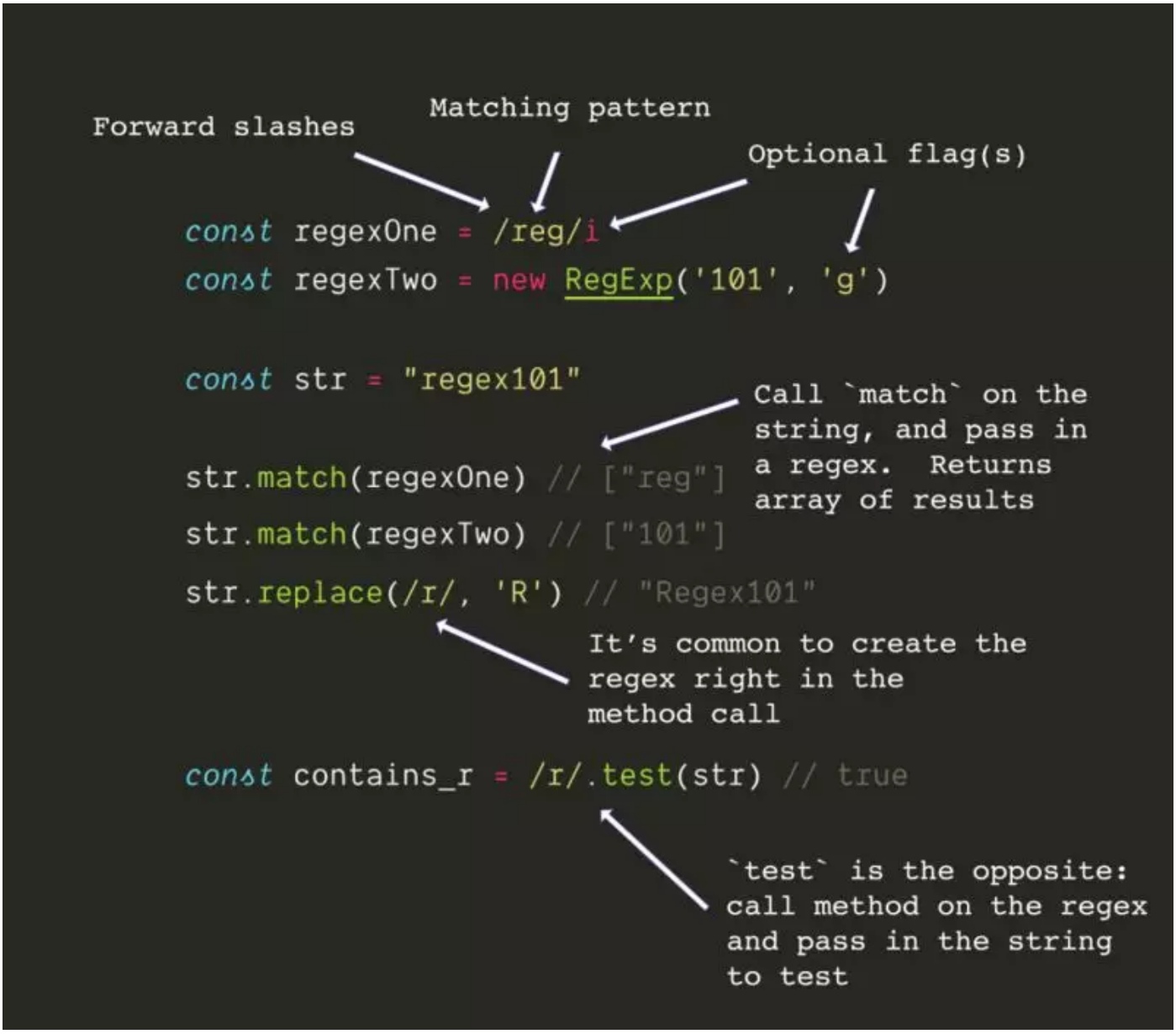
概览

在 JavaScript 中，使用 // 即可创建一个正则表达式对象，当然也可以使用 new RegExp()

常用的跟正则相关的方法有 match、test 和 replace。

其中 match，replace 都是字符串上的方法，test 是正则对象上的方法。

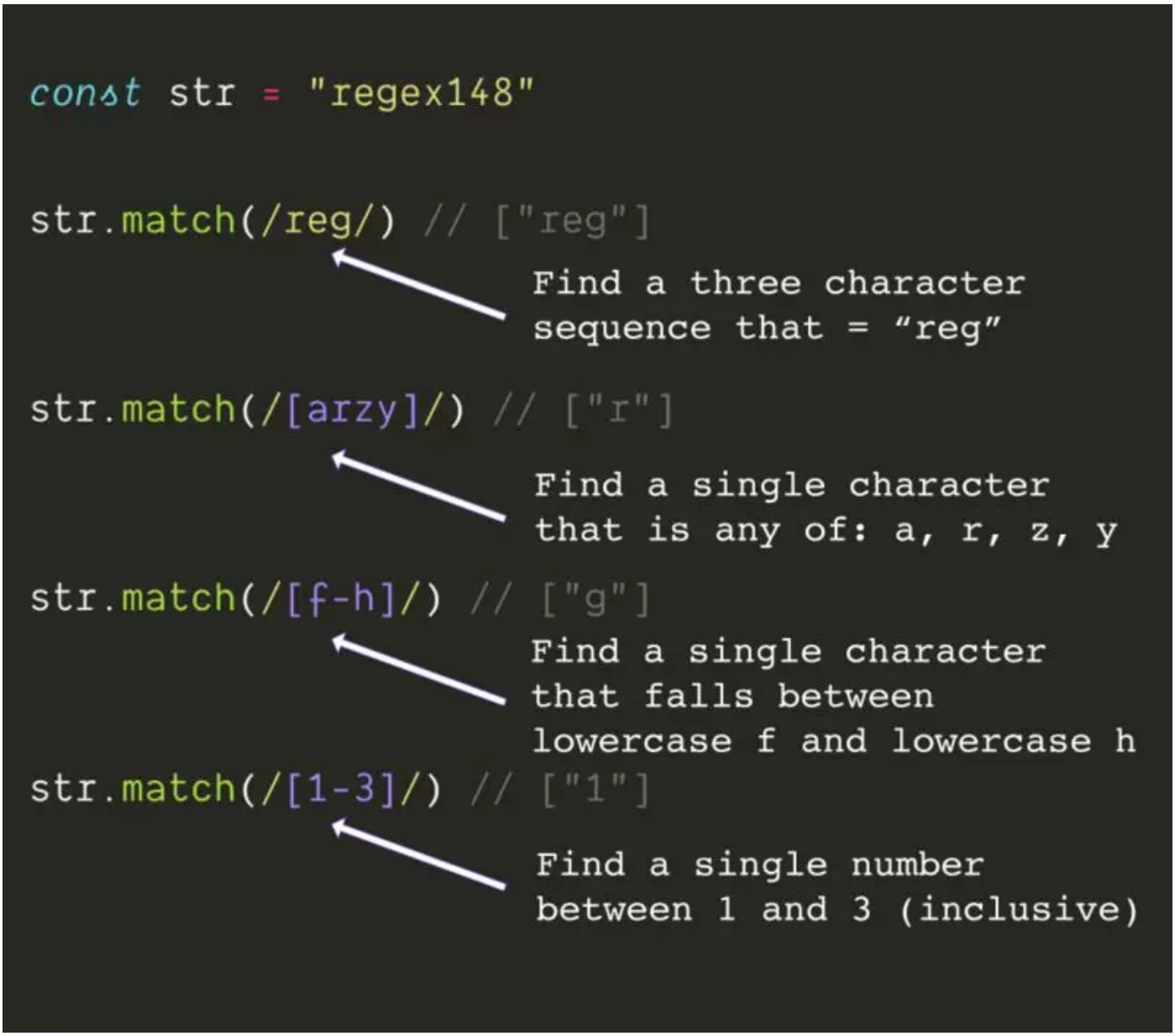
下面看具体的图示：



匹配单字符

- /reg/ 可以直接匹配具体的字符串 reg。

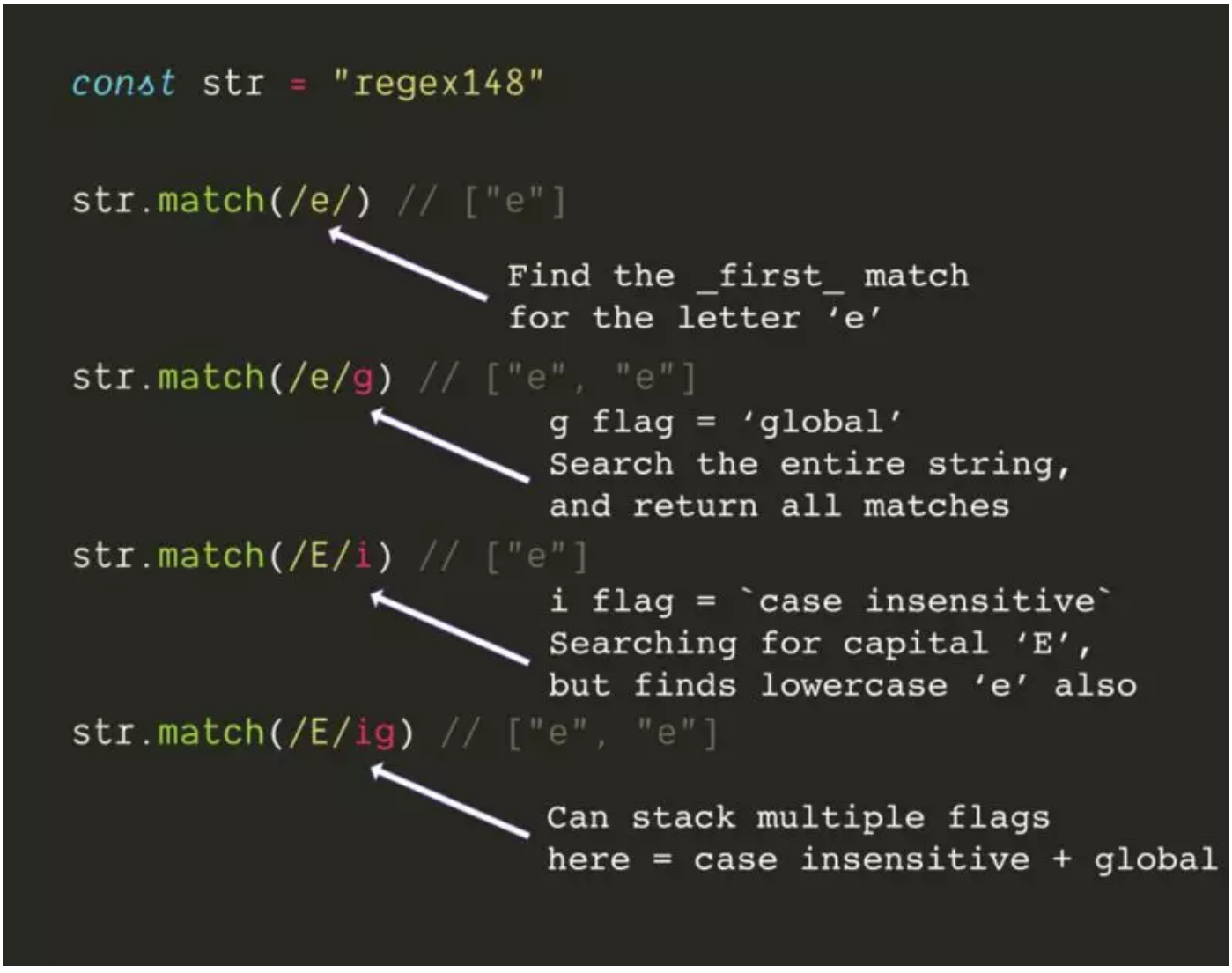
- /[arzy]/ 中的中括号 [] 则代表匹配 arzy 中的任意单个字符
- /[f-h]/ 中的中杠 - 代表匹配字母表顺序中 f 到 h 中的任意单个字符
- /[1-3]/ 则代表匹配数字 1 到 3



正则选项

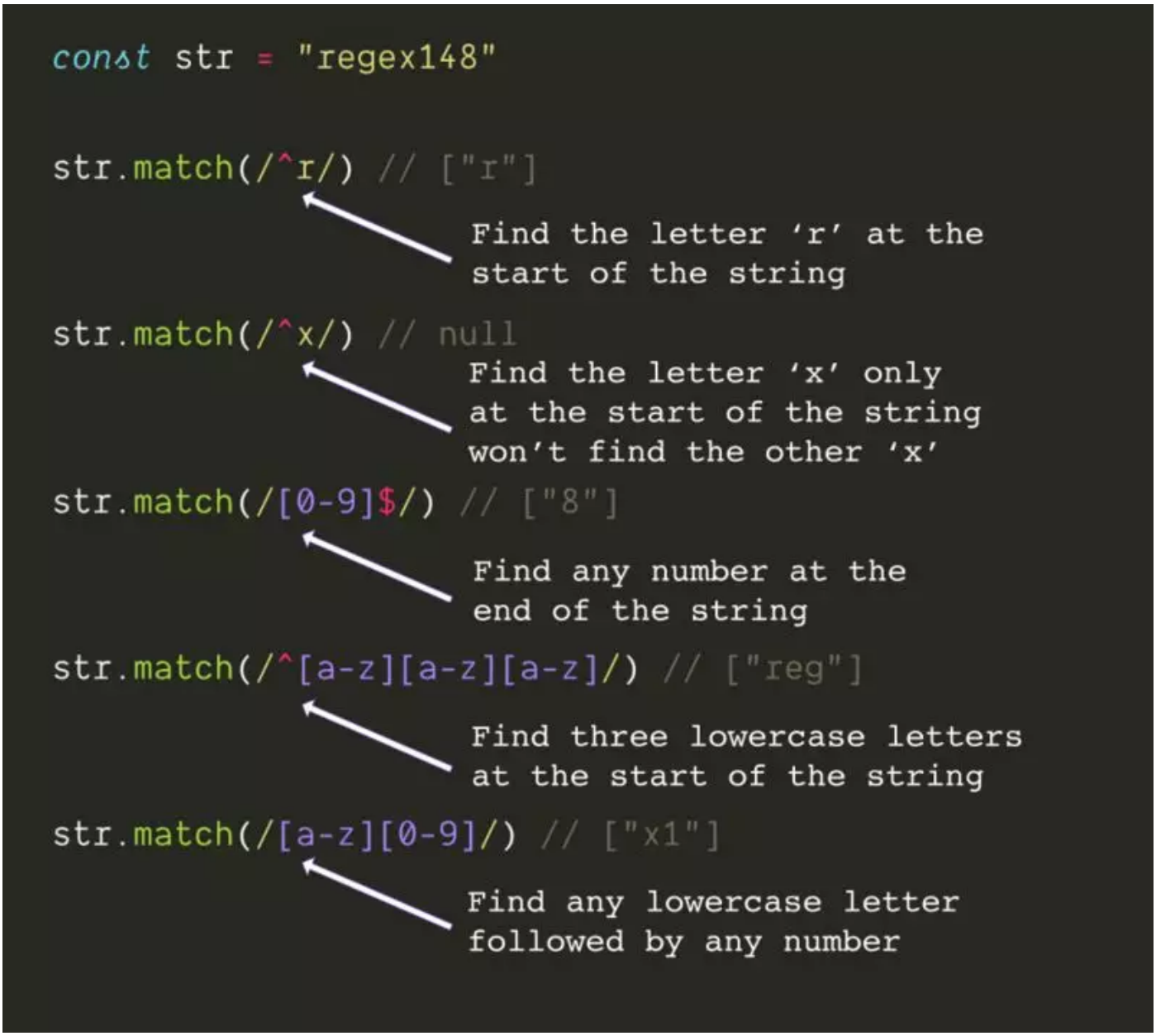
正则对象的后面也可以跟选项，JavaScript 中常用的选项有：

- i - 代表忽略大小写
- m - 代表多行匹配
- g - 代表全局匹配（可以匹配多次）



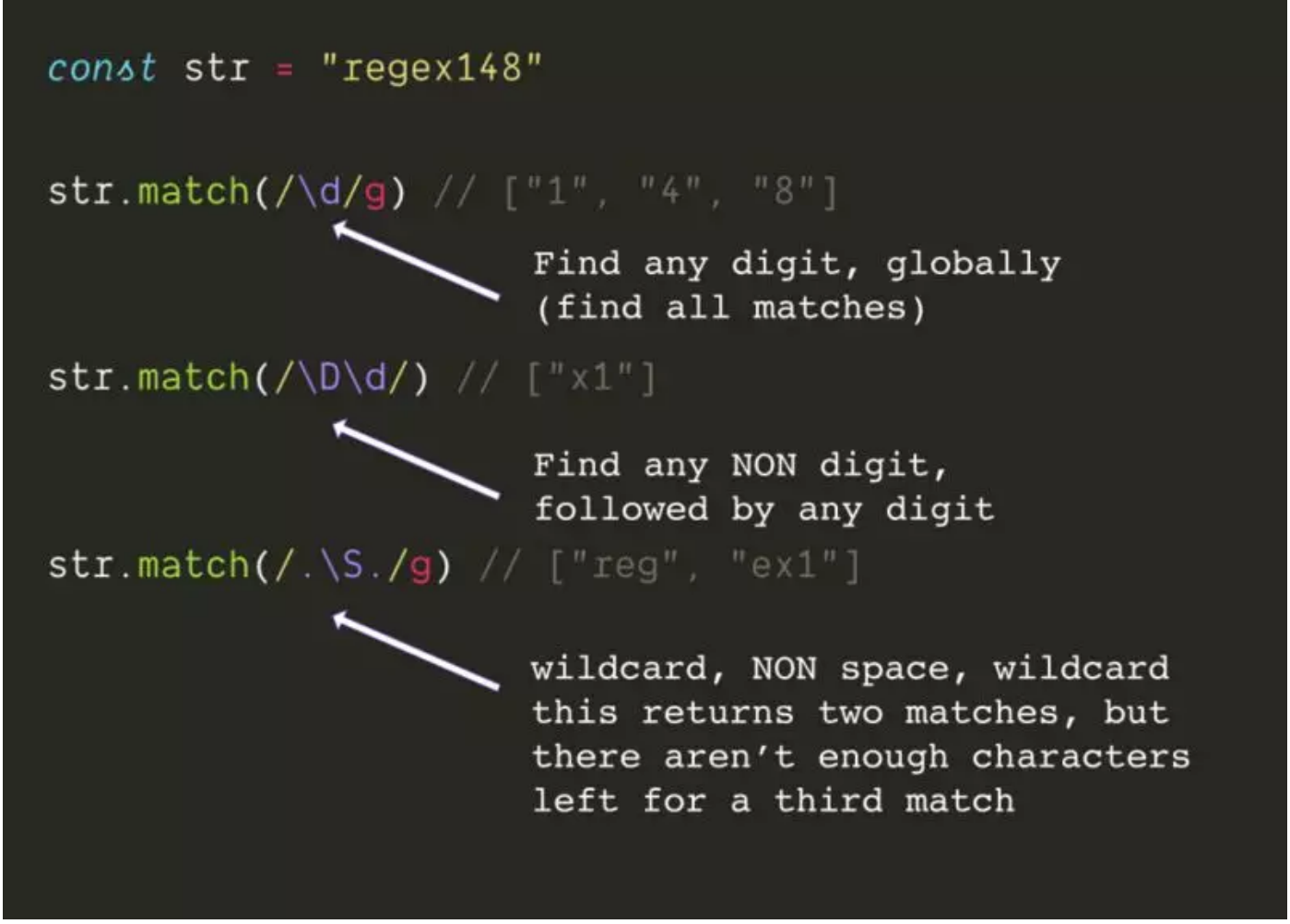
边界匹配

- ^ - 代表匹配字符串的开头
- $ - 代表匹配字符串的结尾



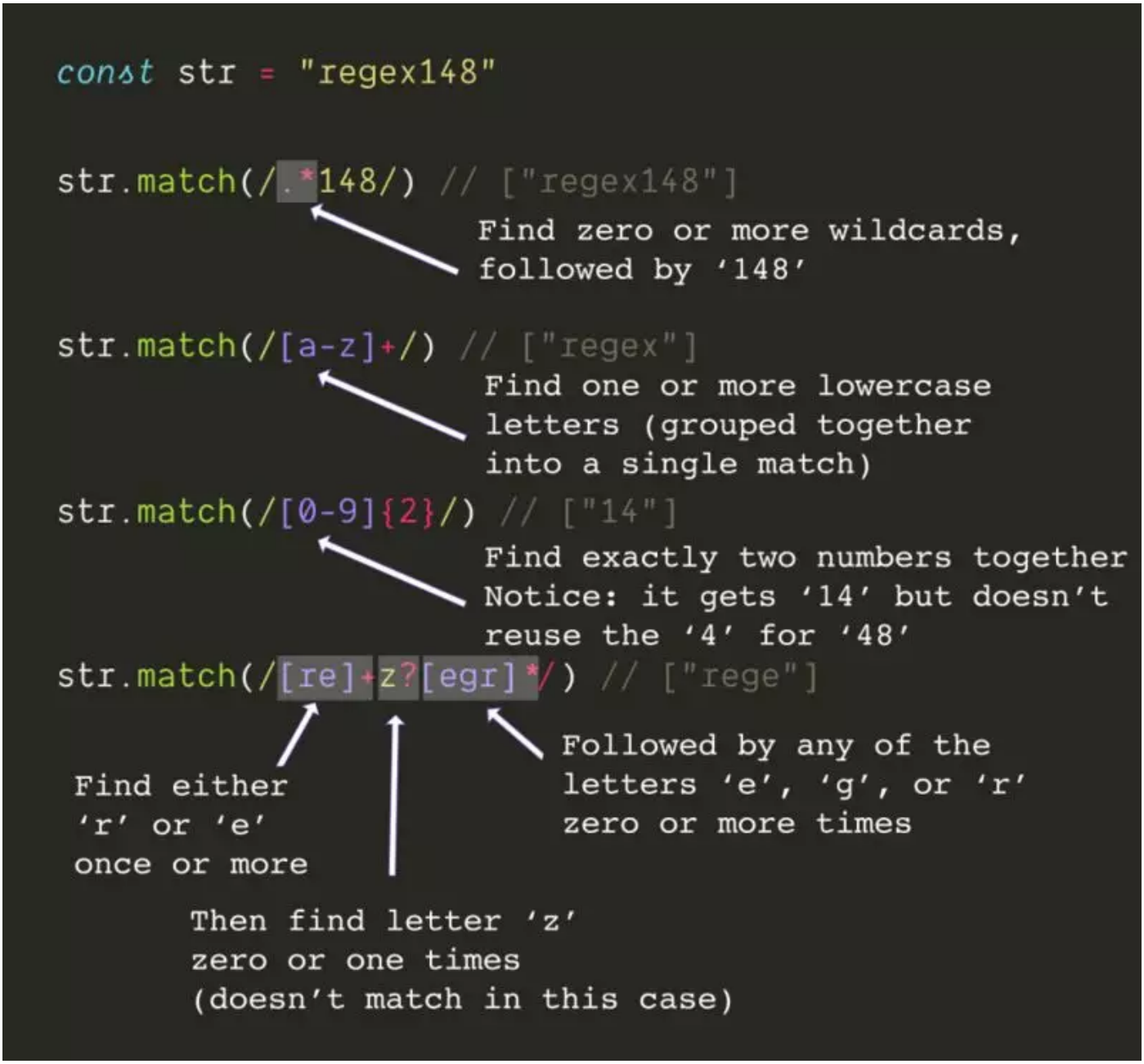
字符匹配

- . - 可以匹配除了换行符外的任意字符
- \d - 可以匹配任意数字
- \D - 可以匹配任意非数字
- \s - 匹配任意空白字符
- \S - 匹配任意非空白字符
- \n - 匹配换行
- \w - 它其实就等同于 [A-Za-z0-9_]，即匹配字母数字下划线



量词匹配

- `*` - 匹配 0 次或多次
- `+` - 匹配 1 次或多次
- `?` - 匹配 0 次或 1 次
- `{3}` - 匹配 3 次
- `{2,4}` - 匹配 2、3 或 4 次
- `{2,}` - 匹配 2 次或多次

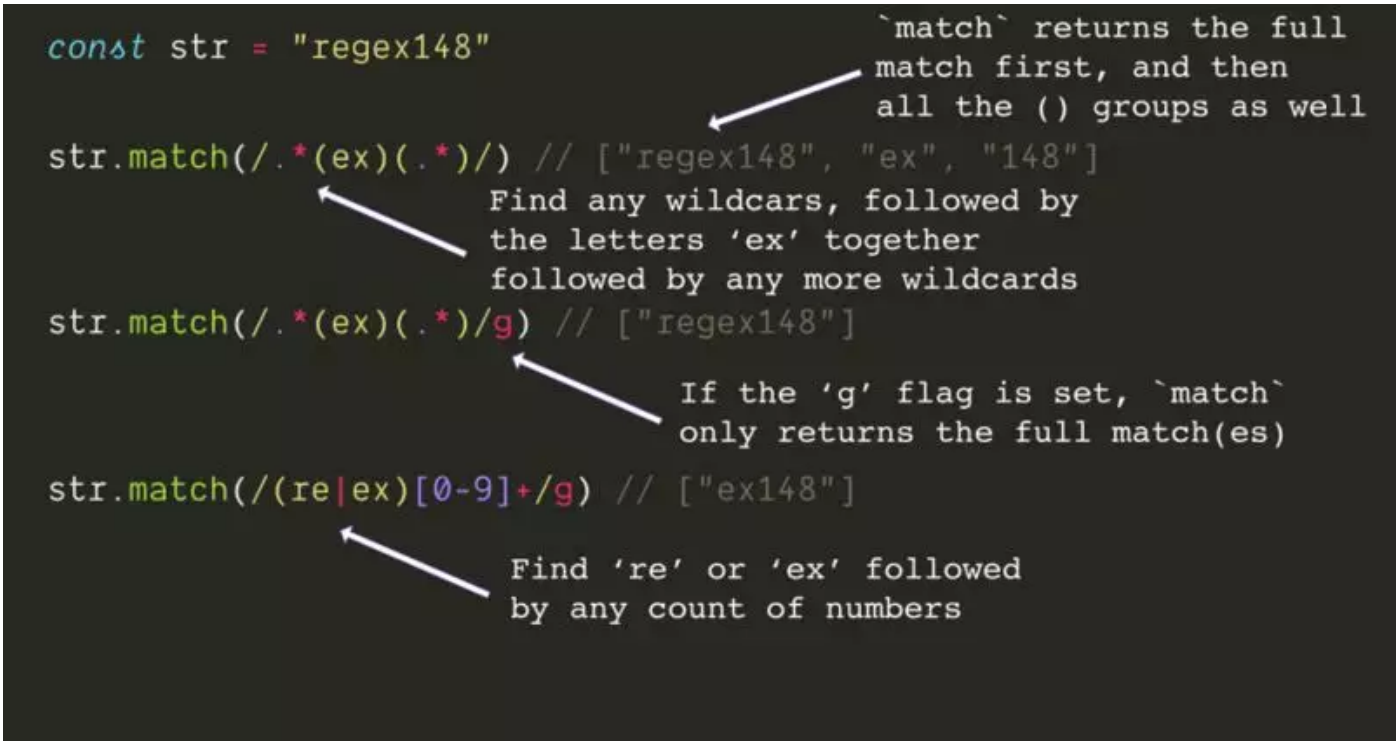


分组

() 小括号在正则中代表分组，一般在 match 方法中用来返回全匹配加上多个分组结果，如果使

用了 g 选项，则只返回全匹配。

在小括号中你可以使用管道符号 |，它代表或

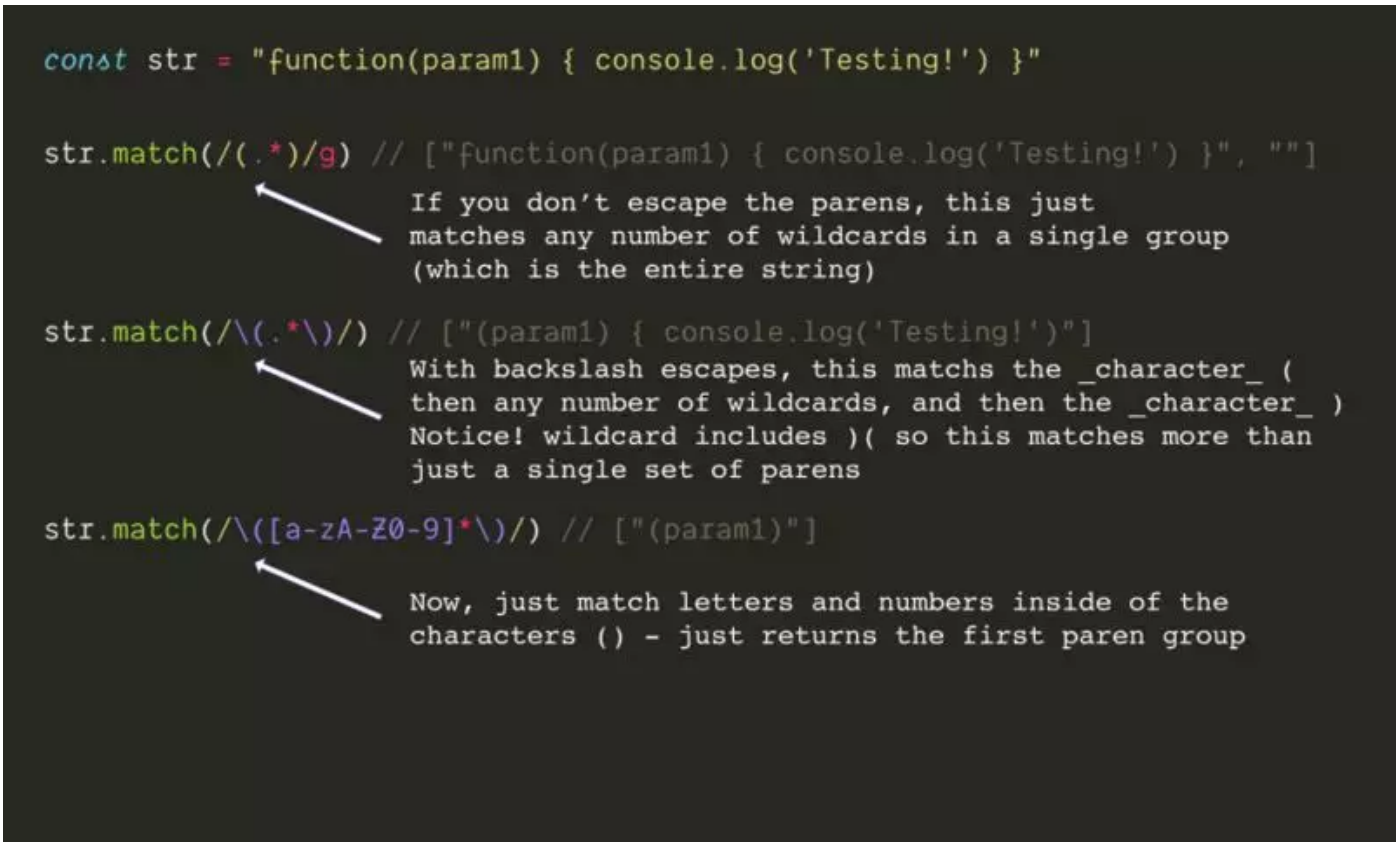


特殊符号

匹配特殊符号的时候需要加反斜杠 \

JS 中的特殊字符有 ^ $ \ . * + ? () [] {} |

所以如果你需要匹配星号 *，就需要这样写：*

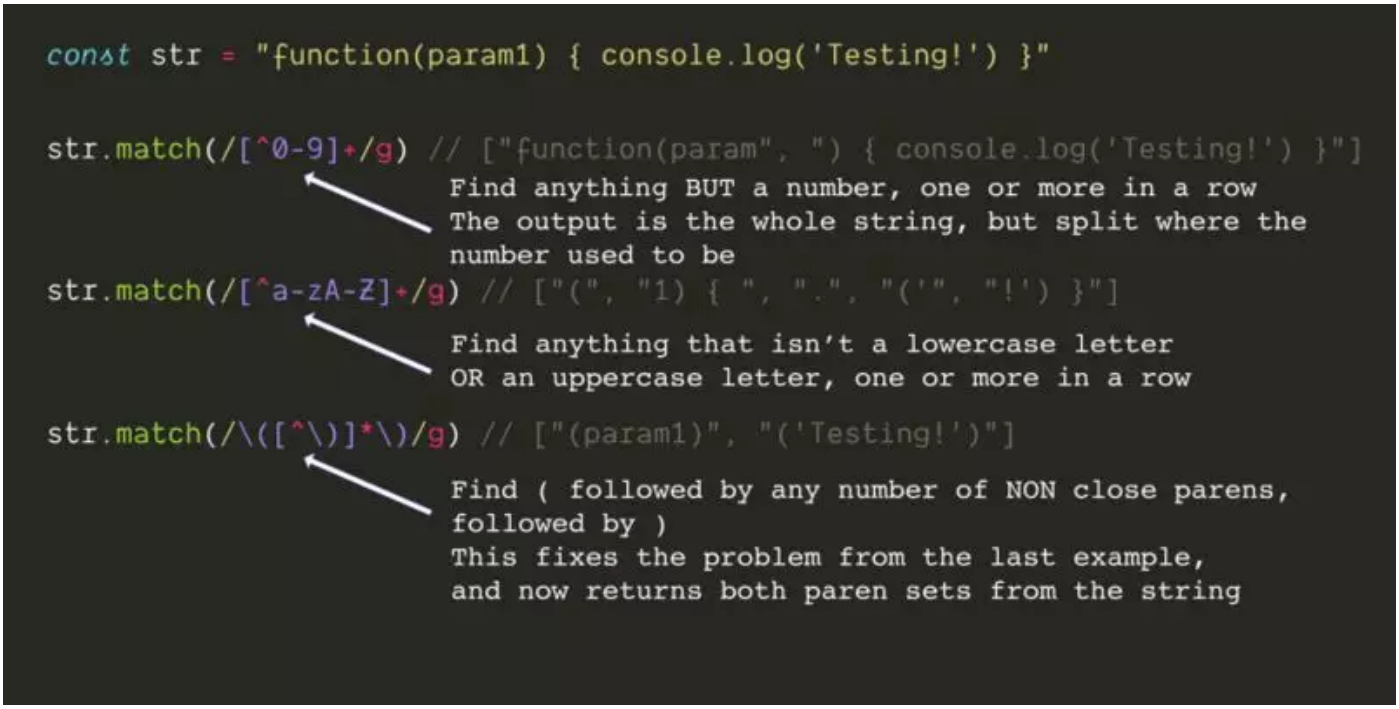


取非匹配

匹配除了某个字符的任意字符，需要在中括号 [] 中使用 ^

至此 ^ 具有两个含义：

- 如果用于正则表达式的开头，代表匹配字符串的开头
- 如果用于中括号 [] 内部，则代表匹配非此字符



结语

上面只是介绍了正则表达式中的基本用法，在日常开发中基本就够用了。更深入的用法比如贪婪

和懒惰、零宽断言和捕获 ，感兴趣的小伙伴可以自行学习~

正则很强大，但是也不能滥用。如果你写出了一个很复杂的只有你能看懂的正则，更好的做法是不使用正则去实现它~

> 参考链接：
> Intro to Regex for Web Developers
>
> SegmentFault 社区链接：
> https://segmentfault.com/a/1190000021145901

- END -

与程序员相处的基本规范

Fasten Your Seatbelts

不会修电脑

不盗QQ号

搭网站请付费

SegmentFault 「思否」
真正属于开发者的社区

长按图片
扫码关注