var str = "sfet33odsufqetu123dsfqwee56abc";

var r = /\d+/g;

var res;

while(res = r.exec(str)){

console.log(res);

}

var str = "sfet33odsufqetu123dsfqwee56abc";

var r = /\d+/g;

1 while(r.exec(str)){

2 console.log(r.exec(str));

}

1. exec()返回的是一个数组，所以要用变量接收，如果不接收，就会陷入死循环，为什么？
2. 运行1行时，exec执行了一次，取到了33；
3. 运行2行时，exec又执行了一次，取到了123并打印出来；
4. 再运行1行时，exec又执行了一次，取到了56；
5. 再运行2行时，exec又执行了一次，取到了null并打印，指针返回；
6. 再运行1行时，exec又执行了一次，取到了33；循环往复。

var str = "sfet33odsufqetu123dsfqwee56abc77";

此时，只打印出123和77，因为此时1行能取到null，就跳出循环了。

-----------------------------------------------------------------------------------------------------------------

var str = "123<div>456</div>789<span>345</span><i></i>abc";

1 var r = /<(\w+)>.\*<\/\1>/g;

2 var r1 = new RegExp("<(\w+)>.\*<\/\1>","g");

var res;

while (res = r1.exec(str)){

console.log(res[0]);

}

1. 第1行是用字面量来创建的正则表达式，第2行是用构造函数来创建的正则表达式，但是当res赋值为r1时打印不出来，为什么？

答：第一种方法中，**/** 有特殊含义，代表正则表达式的声明，所以使用时要用 **\** 转一下；第二种方法，字符串格式可以正常读取**/ \**，此时需要 **\** 来告知哪个是转义字符，所以

var r1 = new RegExp("<(\\w+)>.\*</\\1>","g");

**\**是转义符号，所以在使用字符串表示正则表达式时，\要用双的\\。