

20230919

3.数组

5.数组常用方法

数组头尾删除添加

6.splice方法

利用参数的数量不同实现一个方法具有添加元素，删除元素 更改元素的功能

```
var arr = [1,2,3,4,5,6,7,8,9];

//删除功能 splice（从指定下标元素开始，连续删除几个元素）
// arr.splice(2,3);

//元素更改功能splice（从指定下标元素开始，连续几个元素，变更的目标元素）
// arr.splice(2,3,'a','b');

//增加功能 splice（从指定下标元素开始，0，添加的目标元素）
arr.splice(2,0,'a','b');
console.log(arr);
```

```
D:\Work\zrProject\zrProject\前端\dailyPractice\jsPro\08数组>node demo_06.js
[
  1, 2, 'a', 'b', 3,
  4, 5, 6, 7, 8,
  9
]
```

7.二维数组

数组中的元素仍为数组

```
var citys = [
  [1,2,3],
  [4,5],
  [7,8,9,0],
];

// console.log(citys);

for (var i = 0; i < citys.length; i++){
  for (var j = 0; j < citys[i].length; j++){
    console.log(citys[i][j])
  }
}
```

```
D:\Work\zrProject\zrProject\前端\dailyPractice\jsPro\08数组>node demo_07.js
1
2
3
4
5
7
8
9
0
```

4.String字符串

String: js的内置对象，字符类型的封装类型

```
//字符串的初始化方式/创建字符串
var str = new String('hello world');
console.log(str);
```

```
//字符串的字面量赋值法
var str_2 = 'hello world'
console.log(str_2);
```

```
var str = 'abcde12345';

var str2 = ''
//字符串长度 字符数量
console.log(str.length);
console.log(str2.length);
//根据下标返回下标对应的字符
console.log(str.charAt(2))
//超出范围的下标值返回‘空白’
console.log(str2.charAt(100));
console.log(str.charAt(6)+0);

// charCodeAt 根据下标 返回下标对应的字符的unicode编码
console.log(str.charCodeAt(0));

//indexOf() 根据字符查找下标（返回第一个对应的下标）
console.log(str.indexOf('e'));
console.log(str.indexOf('123'));//进行string查找
console.log(str.indexOf('125'));//不存在则返回-1
// !!!使用场景 indexOf()>来判断是否存在
// lastIndexOf() 从后向前检索字符串
console.log(str.lastIndexOf('1'));
//concat字符串拼接
var str_1 = 'abcde12345'
var str_2 = 'abcde12345'
var str_1 = str_1.concat(str_2)

console.log(str_1);
// slice返回字符串中指定的一部分 slice (a, b) 左闭右开（左侧可以去的下标为a的元素，右侧娶不到下标为b的元素）
//左闭右开格式：【a, b）
var str_3 = 'abcde12345'
str_3 = str_3.slice(2,6);
```

```

console.log(str_3)
//substring 返回从指定位置截取字符串到结束值 同slice截取到指定位置也是左闭右开
var str_4 = 'abcde12345'
var str_4 = str_4.substring(3,5);
console.log(str_4);
//trim 删除字符串前后空格
var str_5 = ' abcde12345 '
str_5 = str_5.trim()
console.log(str_5);
//大小写转换分别是toUpperCase和toLowerCase
var str_6 = 'abcde12345'
str_6 = str_6.toUpperCase();
console.log(str_6)

```

```

var str = 'abcde12345'
//split(x) 使用x作为分隔符 将字符串分割为字符串数组
var arr = str.split('c')

console.log(arr);// 字符c 转化为分隔符号 用来分隔字符串
console.log(str);

var arr_1 = str.split("")
console.log(arr_1);//使用空串来分割 每个字符为一个元素
var arr_1 = str.split('0')
console.log(arr_1);//使用不存在的元素为分隔符 依然转换为数组 数组中只有原字符串1个元素

// 使用案例
//假设有字符串内部是城市数据，提取城市名称
var str_citys = ' 北京，上海，广州 ,深圳 '；
// 将该字符串转化为城市名数组，使用‘，’分割
var arr_citys = str_citys.split(',');
for (var city of arr_citys) {
    console.log(city.trim());
}

```

```

D:\Work\zrProject\zrProject\前端\dailyPractice\jsPro\09字符串>node demo_01.js
[String: 'hello world']
hello world

```

```

D:\Work\zrProject\zrProject\前端\dailyPractice\jsPro\09字符串>node demo_02.js
10
0
c

20
97
4
5
-1
5
abcde12345abcde12345
cde1
de
abcde12345
ABCDE12345

```

```
D:\Work\zrProject\zrProject\前端\dailyPractice\jsPro\09字符串>node demo_03.js
[ 'ab', 'de12345' ]
abcde12345
[
  'a', 'b', 'c', 'd',
  'e', '1', '2', '3',
  '4', '5'
]
[ 'abcde12345' ]
北京
上海
广州
深圳
```

5.Math数学对象

该对象用来执行常见数学运算。

特点：Math对象在网页加载时，已经由js引擎创建出了Math的实例对象

使用Math对象的方式window.Math（window可以省略）

```
console.log(Math.max(5,20))//最大值
console.log(Math.min(5,20))//最小值
console.log(Math.abs(-5))//绝对值
console.log(Math.pow(2,4))//a的b次方
console.log(Math.sqrt(25))//开根号
console.log(Math.PI)//π
console.log(Math.floor(1234.666))//向下取整
console.log(Math.ceil(1234.333))//向上取整
console.log(Math.round(1234.333))//四舍五入
console.log(Math.round(1234.666))
//round如果里面的值是负数则5舍6入
console.log(Math.round(-1234.4))
console.log(Math.round(-1234.5))
console.log(Math.round(-1234.6))
```

```
D:\Work\zrProject\zrProject\前端\dailyPractice\jsPro\10Math数学对象>node demo_01.js
20
5
5
16
5
3.141592653589793
1234
1235
1234
1235
-1234
-1234
-1235
```

random随机数

```
// 0~1的随机数有效位数16位 [0,1)
console.log(Math.random());

var int = Math.floor(Math.random() * 3)+1;
console.log(int);
```

```
D:\Work\zrProject\zrProject\前端\dailyPractice\jsPro\10Math数学对象>node demo_02.js
0.5535253985989139
3
```

6.date日期事件

Date对象:提供用于处理日期和时间的属性和方法

get方法,set方法,其他方法

```
//无参构造 当前系统时间的标准日期时间格式对象
var date = new Date()
console.log(date)

// 有参构造 参数为 从1970年1月1日00:00:00开始计算的毫秒数
var date_2 = new Date(1145141919810)
console.log(date_2)

var date_3 = new Date('2023-01-11 11:45:14')
console.log(date_3)

//静态方法
var date_4 = Date.now()
console.log(date_4)

var date_5 = Date.parse('2023-01-11 11:45:14')
console.log(date_5)
```

```
D:\Work\zrProject\zrProject\前端\dailyPractice\jsPro\11date日期时间>node demo_01.js
2023-09-19T12:40:53.352Z
2006-04-15T22:58:39.810Z
2023-01-11T03:45:14.000Z
1695127253357
1673408714000
```

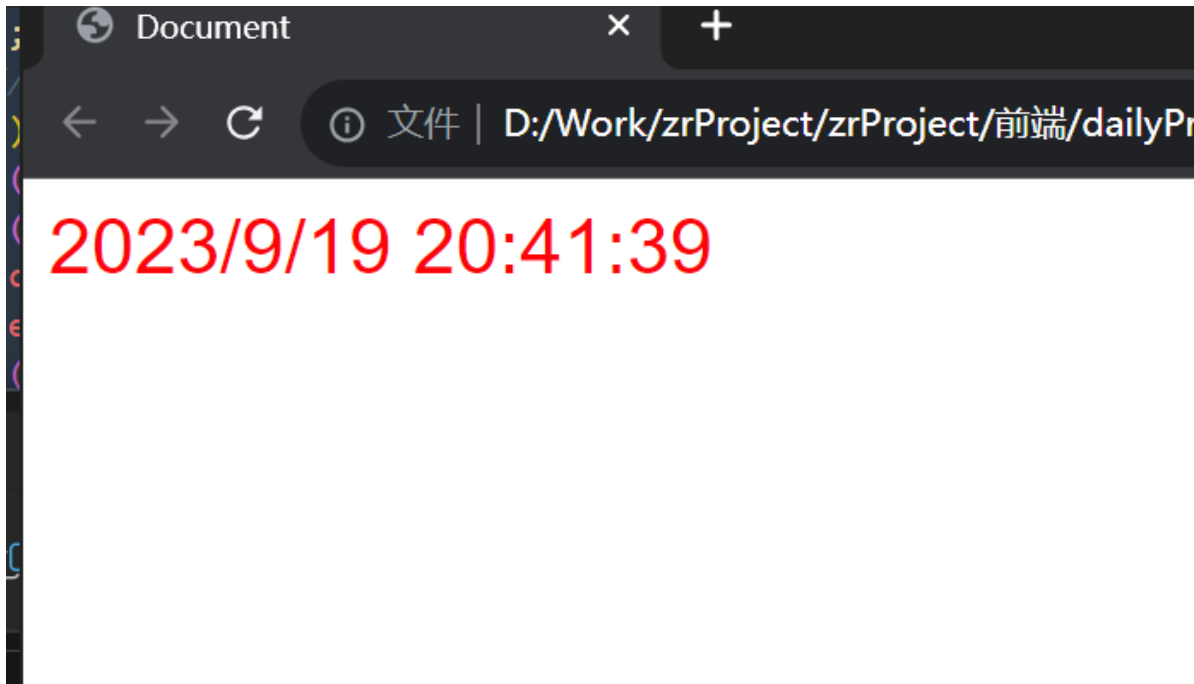
```
var date_1 = new Date();
console.log(date_1.getDate()); //日
console.log(date_1.getDay()); //星期几
console.log(date_1.getHours()); //时
console.log(date_1.getMinutes());
console.log(date_1.getSeconds());
console.log(date_1.getMilliseconds());
console.log(date_1.getTimezoneOffset());
console.log(date_1.getUTCDate());

// 所有能使用get获取的值,可以使用set进行修改设置
date_1.setDate(20)
console.log(date_1.getDate())

console.log(date_1.toString()+'abc')
console.log(date_1.toLocaleDateString()) //常用年月日
console.log(date_1.toLocaleTimeString()) //常用时分秒
console.log(date_1.toLocaleString()) //常用年月日时分秒

function showDate(){
var date_now = new Date();
```

```
date_now = date_now.toLocaleString();
var div01 = document.getElementById('id_div_01');
div01.value=date_now;
}
window.setInterval(showDate,1000)
```



DOM编程

定义:document object model 文档对象模型

作用:访问和处理HTML文档的标准方法,DOM将HTML文档呈现为带有元素属性以及文本的树状结构(节点树)

理解:将html页面上的元素,抽象为js中的一个dom节点(dom节点可理解为dom对象)

1.获取元素对象

1.通过id值获取

格式:document.getElementById

```
var div_01 = document.getElementById('id_div_01')//通过id获取
console.log(div_01)
```

2.通过class获取

```
// var div_02 = document.getElementsByClassName('class_div_01')
var div_class = document.getElementsByClassName('class_div_01')[0]
// console.log(div_02)
console.log(div_class)
```

☑ Show CORS errors in console

```
<div id="id_div_01">第一个div</div>
```

demo_01.js:2

```
<div class="class_div_01">第二个div</div>
```

demo_01.js:7

```
> |
```

作业

1.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>demo_01</title>
</head>
<body>
  <input type="text" id="inputString" placeholder="请输入不少于三个数的字符串">
  <button onclick="getThreeNumber()">Touch me</button>
  <script>

    function getThreeNumber() {

      var number = document.getElementById("inputString").value;

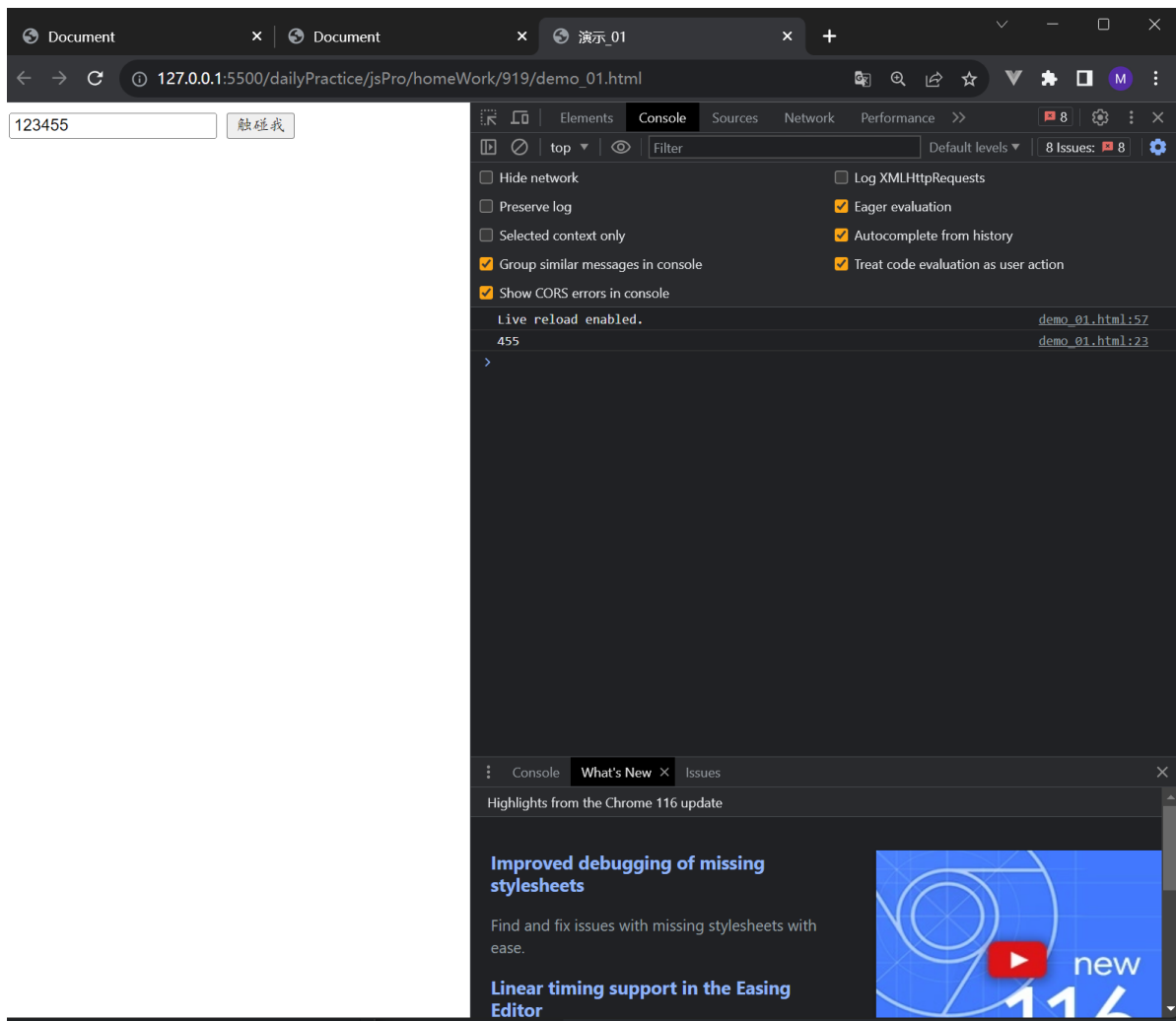
      if (number.length < 3){

        console.log("请输入3个以上字符")
      }else{

        var lastThree = number.substring(number.length - 3);
        console.log(lastThree)
      }

    }

  </script>
</body>
</html>
```



2.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>demo_01</title>
</head>
<body>
  <input type="text" id="inputString" placeholder="请输入身份证号">
  <button onclick="getIDNumber()">Touch me</button>
  <script>

    function getIDNumber() {

      var number = document.getElementById("inputString").value;

      if (number.length != 18){

        console.log("请输入正确格式的身份证")
      }else{

        var year = number.substring(6,10);
        var month = number.substring(11,12);
        var day = number.substring(13,14);
        console.log('生日是'+year + "年" + month + "月" + day+ "日")
      }
    }
  </script>
</body>
</html>
```

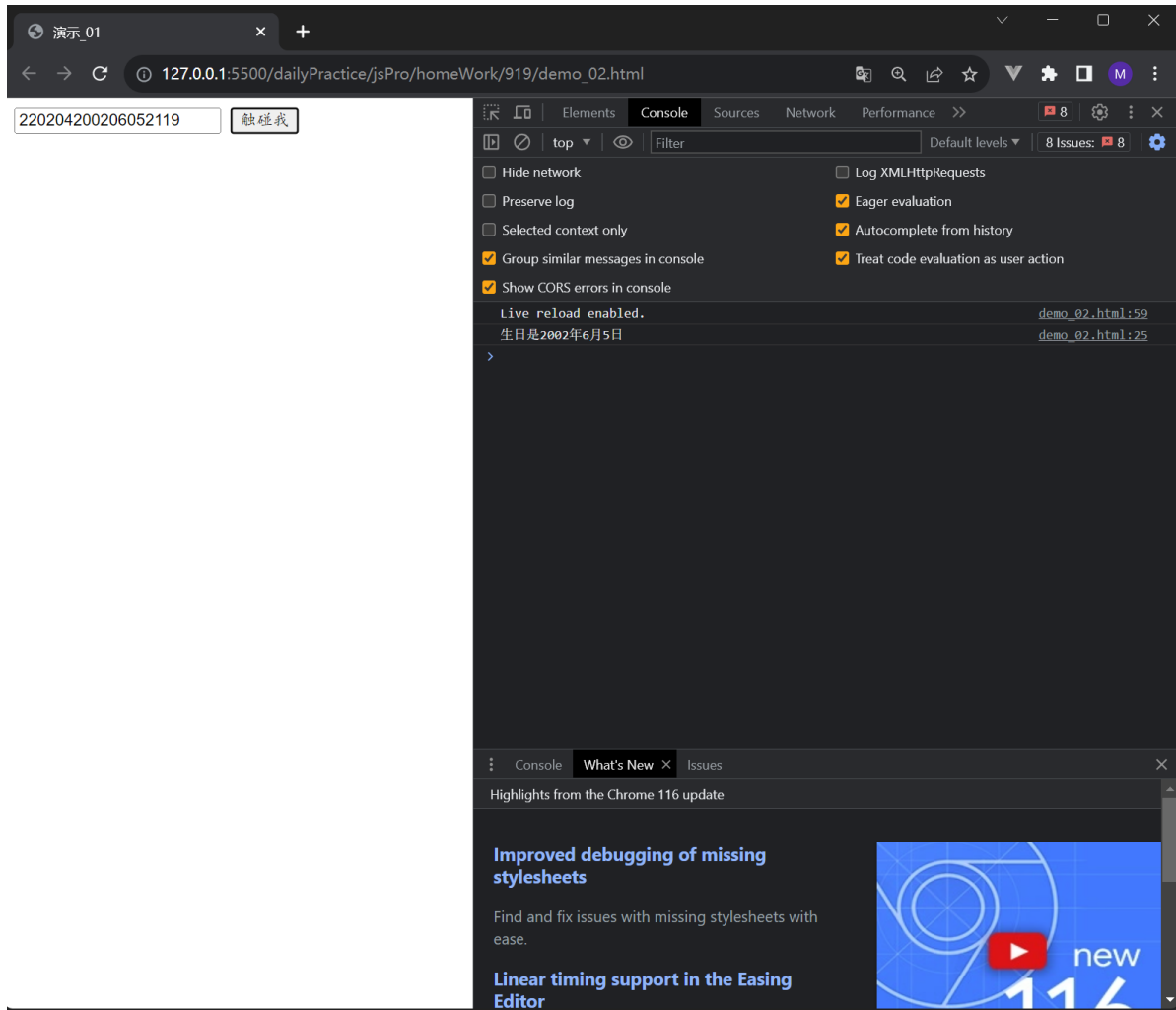


```

    }

</script>
</body>
</html>

```



3.

```

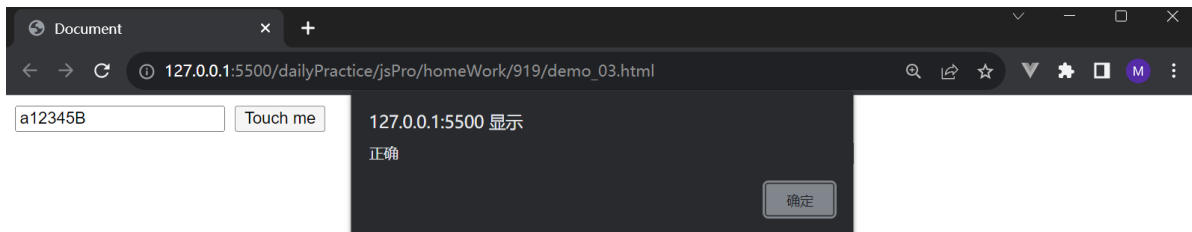
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <input type="text" id="inputString" placeholder="请输入用户名">
  <button onclick="judge()">Touch me</button>

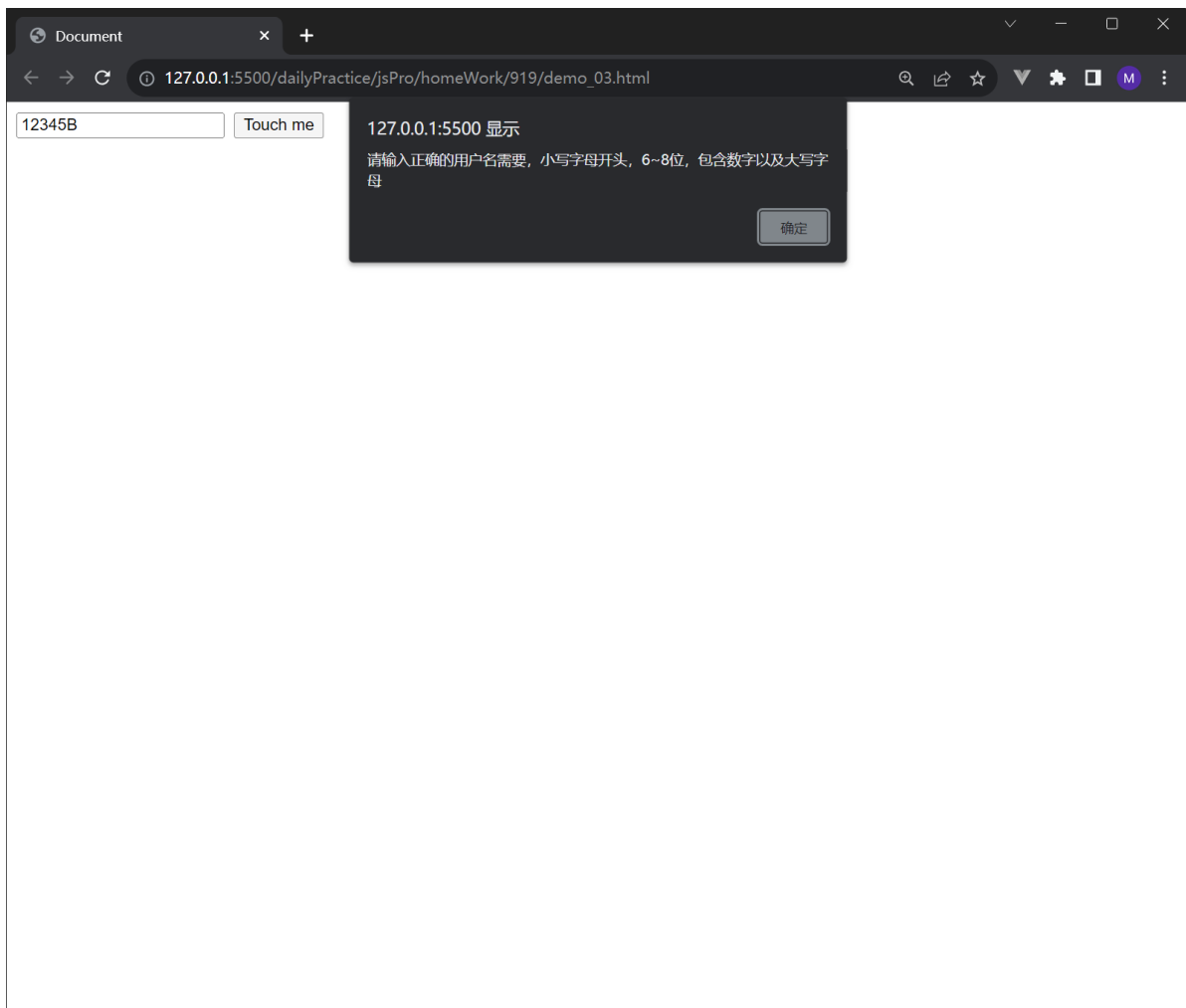
  <script>
    function judge() {
      var inputString = document.getElementById("inputString").value;

```

```
        if (inputString.length >= 6 && inputString.length <= 8 &&
inputString[0] >= 'a' && inputString[0] <= 'z') {

            for (var i = 0; i < inputString.length; i++) {
                if (!(inputString[i] >= 'A' && inputString[i] <= 'Z' ||
inputString[i] >= '0' && inputString[i] <= '9')) {
                    alert('正确')
                    break;
                }
            }
        }
    }
    else {
        alert('请输入正确的用户名需要，小写字母开头，6~8位，包含数字以及大写字母');
    }
}
</script>
</body>
</html>
```





4.

```
<!DOCTYPE html>
<html>
<head>
  <title>生成验证码</title>
</head>
<body>
  <div id="captcha"></div>
  <button onclick="generateCaptcha()">刷新验证码</button>

  <script>
    function generateCaptcha() {
      var storage =
'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789';
      let captcha = '';
      for (let i = 0; i < 4; i++) {
        var randomIndex = Math.floor(Math.random() * storage.length);
        captcha = captcha+ storage[randomIndex];
      }
      document.getElementById('captcha').textContent = captcha;
    }
    generateCaptcha();
  </script>
</body>
</html>
```

hFhD

刷新验证码

xn40

刷新验证码

5.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <input type="text" placeholder="请输入初始日期 格式: xxxx/xx/xx" id="id_inp_1"
style="width: 250px"><br>
  <input type="text" placeholder="请输入结束日期 格式: xxxx/xx/xx" id="id_inp_2"
style="width: 250px"><br>
  <button onclick="calculate()">Touch</button>

  <script>

    function calculate() {

      var inp01 = document.getElementById("id_inp_1").value;
      var inp02 = document.getElementById("id_inp_2").value;

      var date01 = new Date(inp01);
      var date02 = new Date(inp02);
      var date03 = new Date();
      date03=date02-date01
      var a = 24*60*60*1000;
      alert('相差了'+date03/a+'天');

    }

  </script>
</body>
</html>
```

