JS第二周第一天

`javascript`

1.循环绑定事件

```
<body>
<ul id="list">
    <li>1</li>
    <li>2</li>
    <li>3</li>
    <li>4</li>
    <li>5</li>
</ul>
</body>

<script>
    let list=document.getElementById("list");
    let oLis=list.getElementsByTagName("li");
    for (var i=0;i<oLis.length;i++) {
      debugger;
      //console.log(i);
      oLis[i].a=i;
      console.log(oLis[i].a);
      oLis[i].onclick=function () {
        //只有用户点击的时候才会执行  当点击的时候代码(循环)早就执行完了
        //此时的i 5
        //debugger;
        console.log(i);
        console.log(this.a);
      }
    }
    //debugger;
    console.log(i);//5
    console.log(oLis);

    //先执行循环再执行事件绑定的函数, 执行事件绑定的函数的时候循环早就结束了,所以在事件绑定的函数中的i是循环结束之后的i
    //如果你想在绑定事件的函数中拿到对应li的索引不可以使用i
    //我们可以使用自定义属性 来将他的索引存起来 到时候可以通过this.自定义属性获取出来
</script>
```

2.循环绑定事件的例子

```
<style>
        .box {
            width: 100px;
            height: 100px;
            border: 1px solid red;
            margin: 10px;
            display: inline-block;
            background-color: yellowgreen;
        }

        .box.box1 {
            background-color: #fff;
        }
    </style>

<body>
<div class="box"></div>
<div class="box"></div>
<div class="box"></div>
<div class="box"></div>
<div class="box"></div>
<div class="box"></div>
<div class="box"></div>
<div class="box"></div>
<div class="box"></div>
<div class="box"></div>
</body>

<script>
  let boxList = document.getElementsByClassName("box");
  for (var i = 0; i < boxList.length; i++) {
    boxList[i].f=true;
    boxList[i].onclick=function () {
      this.f=!this.f;
      //"box"  =>"box box1"
      //this.className="box box1";
     // this.className+=" box1"; 不要忘了 box1前面加上一个空格
      //根据自己的自定义属性f的值来确定自己的className是"box"还是"box box1"
      if(this.f){
        this.className="box";
      }else {
        this.className="box box1";
      }
    }
  }
  console.log(boxList);
</script>
```

3.拼接字符串绑定到页面上

```
<style>
        ul,li{
            list-style: none;
        }
        li img{
            width: 100px;
            height: 100px;
        }
        li p{
            display: inline-block;
        }
    </style>

<body>
<ul id="list"></ul>
</body>

<script>
  let data = [{img: "1.jpg", text: "我是第一张图"}, {img: "2.jpg", text: "我是第二张图"}, {img: "3.jpg", text: "我是第三张图"}, {img: "4.jpg", text: "我是第四张图"}];
  let list=document.getElementById("list");
  let str=``;
  //循环数组data 拼接li
  for (var item of data){
      //item 是数组data的每一项 是一个对象想要获取对象中的属性值使用 item.im
g 和item.text
      //既然是拼接字符串 就是在原来的基础上多加一个 使用 str+=
      //使用ES6模板字符串 ${}变量
      str+=`
      <li>
        <img src="img/${item.img}" alt="">
        <p>${item.text}</p>
      </li>
      `
    }
  //将拼接好的字符串放在页面上 就是给list.innerHTML
  list.innerHTML=str;
</script>
```

4.字符串

```
<script>
    let str="ASDASDFFGFAS";
    console.log(str.__proto__);
    //1.str.charAt(索引) 跟 str[索引]一样用
    console.log(str.charAt(1));

    //2.indexOf  lastIndexOf
    //str.indexOf() 判断某一个字符在字符串中首次出现的索引
    //没有就是-1
    console.log(str.indexOf("F"));
    console.log(str.indexOf("f"));//-1
    //参数是一串字符串 先看这个字符串在原字符串中有没有 有的话再看首字符在原字符中的索引
    console.log(str.indexOf("ASD"));//0

    //str.lastIndexOf() 最后一次出现的索引  没有也是-1;
    str.lastIndexOf("G");

    //3.includes 判断某字符在不在指定字符串中 在就是true,不在就是false
    console.log(str.includes("S"));//true
    console.log(str.includes("s"));//false

    //4.split(指定字符)  将字符串按照指定字符拆分为数组
    let str1="2018-4-24";
    console.log(str1.split("-"));//["2018","4","24"]
    console.log("12345".split(""));//["1","2","3","4","5"]

    //5.replace(字符1,字符2),将字符1替换成字符2  replace一次只能替换一个
    console.log(str1.replace("-", "/"));
    //6.slice(n,m)查找 从索引n找到索引m不包括m ,如果m超出索引的最大值也是找最后即可
    //slice(n):参数只有一个n,就是从索引n找到最后
    //slice():没有参数默认就是 slice(0)
    //参数是负数:倒着数(最后一个是-1),正着找
    let str2="ASDFGH";
    console.log(str2.slice(2, 4));//"DF"
    console.log(str2.slice(2, 10));//"DFGH"
    console.log(str2.slice(1));//"SDFGH"
    console.log(str2.slice());//"ASDFGH"
    console.log(str2.slice(-3));//"FGH"
    console.log(str2.slice(-2));//"GH"

    //7.subStr  subString
    //subStr(n,m)从索引n找m个 如果m超出范围也是找到最后即可
    //subStr(n)从索引n找到最后
    //负值也是一样的倒着数即可
    console.log(str2.substr(2, 4));//"DFGH"
    console.log(str2.substr(2, 10));//"DFGH"
    console.log(str2.substr(1));//"DFGH"
    console.log(str2.substr(-3));//"FGH"
    //substring(n,m)从索引n到索引m
    //跟slice意思一样只是不识别负数 负数识别为0
    console.log(str2.substring(-3));

    //8. str2.toLowerCase()  转为小写   str2.toUpperCase()  转为大写
    console.log("ASD".toLowerCase());

    //9.repeat(数字) 将字符串复制几次
    //参数:取整
    //负数会报错
    //如果是-零点几就是0  结果就是""
    console.log("ASD".repeat(3));//"ASDASDASD"
    console.log("ASD".repeat(-0.1));//""

    //10.startsWith(指定字符,第二个参数) 是不是以指定字符作为开头
    //第二个参数 可选的,传了就表示开始查找的位置,不传默认就是0

    //endsWith(指定字符,第二个参数)是不是以指定字符作为结尾
    console.log("ASD".startsWith("A"));//true
    console.log("ASD".startsWith("A",1));//false
    console.log("ASD".startsWith("S",1));//true

    //11.padStart(n,"指定字符")
    //将得到新字符串的length变成n,使用指定字符在前面补位
    //padEnd(n,"指定字符")
    //将得到新字符串的length变成n,使用指定字符在后面补位

    //"指定字符有多个如果超出会自动截取
    console.log("asd".padStart( 10,"A"));
    console.log("asd".padStart( 10,"AB"));//"ABABABAasd"
</script>
```

5.while循环

```
<script>
    //while (条件) {循环体}
    for (var i=0;i<5;i++){}

    var i=0;
    while (i<5){
      //debugger;
      i++;
      console.log("哈哈",i);
    }

    let arr=[1,2,12,23,"11",12,1];
    //找出数组中的唯一一个字符串的索引

    for (var i=0;i<arr.length;i++){
      if(typeof arr[i]=="string"){
        console.log(i);
        break;
      }
    }

    //什么情况下继续找
    //不是字符串的时候继续找
    var i=0;
    while (typeof arr[i]!="string"){
      i++;
    }
    console.log(i);

    //全部替换
    let str="1+2+3+4+5+6+7+8";
    //将+全部替换成*
    // str=str.replace("+","*").replace("+","*");
    // console.log(str);

    //不确定替换多少次  使用while
    //满足什么样的条件我们继续替换
    //只要还有+ 就继续替换
    while (str.includes("+")){
      str=str.replace("+","*");
    }
    console.log(str);


    String.prototype.includes=function(str1) {
      let f=false;
      for (let i=0;i<this.length;i++){
        if(this[i]==str1){
          f=true;
          break
        }
      }
      return f;
    }
</script>
```

6.字符串练习

```
<script>
    //1.var str=”12345678” ;
    //得到一个数组[1,2,3,4,5,6,7,8]
    //注意:数组中每一项都是数字不是字符串

    var str="12345678";
    //console.log(str.split(""));//["1","2",,,,]

    var ary=[];
    for (var i=0;i<str.length;i++){
      ary[i]=Number(str[i])
    }
    console.log(ary);


    var str="www.zhufengpeixun.com?a=1&b=2&c=3";
    // 1)将上面字符串得到一个数组[“a=1”,”b=2”,”c=3”];
    //这个字符串我们只需要?后面的部分
    //将字符串使用split按照?拆分为数组,获取数组的后面那一项
    str=str.split("?")[1];//"a=1&b=2&c=3"
    let ary1=str.split("&");
    console.log(ary1);//["a=1", "b=2", "c=3"]

    // 2)将上面字符串得到2个数组[“a”,”b”,”c”]和[“1”,”2”,”3”];
    //["a=1", "b=2", "c=3"] 将这个数组每一项按照=拆分 第一项放在一个数组中第二项放在一个数组中
    let arr1=[],arr2=[];
    for (let item of ary1){
      arr1.push(item.split("=")[0]);
      arr2.push(item.split("=")[1]);
    }
    console.log(arr1,arr2);

    // 3)将上面字符串得到一个对象{a:1,b:2,c:3};
    //["a=1", "b=2", "c=3"] ==> {a:1,b:2,c:3}

    let obj={};
    for (let item of ary1){
      obj[item.split("=")[0]]=item.split("=")[1];
    }
    console.log(obj);



//     4.var str=”asdfaaassdddfffssss”
//     1)找出出现次数最多的字符并求出他的次数
//     2)得出每个字符出现的次数(可以放在一个对象中)
//     3)得出去除重复后的字符串
    var str="asdfaaassdddfffssss";
    let objNum={};
    for (let i=0;i<str.length;i++){
      if(str[i] in objNum){
        objNum[str[i]]+=1;
      }else {
        objNum[str[i]]=1;
      }
    }
    console.log(objNum);
    //{a: 4, s: 7, d: 4, f: 4}

    //{a: 4, s: 7, d: 4, f: 4} 求出属性值最大的属性名和属性值
    //假设法
    //假设最大次数是0  max=0;  对应的字符是 s="";

    let max=0,s="";
    for (let key in objNum){
      if(objNum[key]>max){
        max=objNum[key];
        s=key;
      }
    }
    console.log("出现次数最多的字符是:"+s+" 次数是:"+max);

</script>
```

7.Math函数

```
<script>
  //最大最小值
  Math.max(1, 2, 3, 4);//4
  Math.min(1, 8, 9);//1

  //向上取整
  Math.ceil(1.1);//2
  //向下取整
  Math.floor(1.9);//1
  //四舍五入
  Math.round(3.7);//4

  //Math.pow(x,y) x的y次幂
  console.log(Math.pow(2, 3));
  //算数平方根
  Math.sqrt(9);//3

  //abs 绝对值  |-2|
  console.log(Math.abs(-4));

  console.log(Math);

  //随机数
  Math.random();//0-1的随机小数
  //n-m的随机数
  Math.random() * (m - n) + n;

  //n-m的随机整数
  //Math.round(Math.random() * (m - n) + n);

  //Math.PI   π

</script>
```

8.获取验证码

```
<style>
        #count {
            width: 200px;
            height: 80px;
            border: 1px solid red;
            margin: 50px auto 10px;
            background: -webkit-linear-gradient(top left, red, gol
d, blue, green, yellow);
            color: white;
            font-size: 50px;
            line-height: 80px;
            text-align: center;
            font-style: italic;
        }

        #change {
            margin: 0px auto;
            text-align: center;
            font-size: 20px;
            cursor: pointer;
            -webkit-user-select: none;
        }
    </style>


<body>
<div id="count"></div>
<p id="change">看不清换一个</p>
</body>

<script>
  let str = "0123456789" + "qwertyuiopasdfghjklzxcvbnm" + "qwertyui
opasdfghjklzxcvbnm".toUpperCase();
  let count = document.getElementById("count");
  let change = document.getElementById("change");
  //随机获取一个 怎么随机?什么随机
  //0-61  只需要随机获取一个0-61的数作为索引
  //获取四位验证码 那就循环4次 获取一个拼接一个
  //str[索引]
  function getCode() {
    let s = "", n = null;
    for (let i = 0; i < 4; i++) {
      //获取随机数
      n = Math.round(Math.random() * (61 - 0) + 0);
      s += str[n];
    }
    //将验证码放在 页面的count中
    count.innerHTML = s;
  }

  //先让函数执行一次 初始的时候才有验证码
  getCode();
  //给change绑定点击事件
  change.onclick = getCode;
</script>
```

9.获取四位不同的验证码

```
<script>
  let str = "0123456789" + "qwertyuiopasdfghjklzxcvbnm" + "qwertyuiopasdfghjklzxcvbnm".toUpperCase();
  let count = document.getElementById("count");
  let change = document.getElementById("change");

  let s="",n=null;
  while (s.length<4){
    n=Math.round(Math.random()*61);
    //判断一下 s中有没有随机获取的字符 str[n],没有才放进去
    if(!s.includes(str[n])){
      s+=str[n];
    }
  }
  count.innerHTML=s;
</script>
```

10.字符串中注意的问题

```
<script>
    let arr=[1,2,3];
    arr[1]=100;
    arr[3]=400;
    arr;//[1,100,3,400];
    arr.length--;
    console.log(arr);

    let str="123";
    str[1]="h";
    console.log(str);
    str[3]="4";
    console.log(str);
    str.length--;
    console.log(str);

    //不可以改变原字符串 除非 直接str="1111",给变量重新赋值
    console.log(Object.getOwnPropertyDescriptor( str,"length"));
    console.log(Object.getOwnPropertyDescriptor( arr,"length"));

    //字符串中所有的方法都不会改变原字符串
</script>
```