JS第二周第三天

1.数组

```
<script>
    Number();
    Boolean();
    Array();//将参数变成数组
    console.log(Array(1, 2, 3));
    //如果参数是只有一个 并且是一个数字 代表得到数组的length
    console.log(Array(7));//[,,,,,,,]

    //面试题:得到一个数组有7个1
    Array(7).fill(1);
    //[[[[[[[1,2],[3,4]]]]]]]  -> "1,2,3,4"
    console.log([[[[[[[1, 2], [3, 4]]]]]]].toString());//1,2,3,4
    console.log([[[[[[[1, 2], [3, 4]]]]]]].join());//1,2,3,4
    console.log([[[[[[[1, 2], [3, 4]]]]]]]+"");//1,2,3,4

    Array.of();//跟Array方法一样 唯一不同的是解决了 只有一个参数是数字的情况
    console.log(Array.of(7));//[7]
    console.log(Array("哈哈"));//["哈哈"]


    //Array.from()
    //参数
    // 数组:得到一个一样的数组
    // 类数组:返回一个数组    将类数组变成数组

    function sum() {
      let total=null;
      for(let i=0;i<arguments.length;i++){
        total+=arguments[i];
      }
```

```
        return total;
      }

      function sum(){
        //Array.from(arguments)  将arguments变成数组,
        //变成数组之后就可以使用数组中的方法 join("+") ->"1+2+3+4+5+6+7+8"
        //使用eval将字符串"1+2+3+4+5+6+7+8" 变成表达式 1+2+3+4+5+6+7+8
        //返回结果 return
        return eval(Array.from(arguments).join("+"));
      }

      console.log(sum(1, 2, 3, 4, 5, 6, 7, 8));

      let ary=[1,2,3,4,5,6,7,8];
      let total=eval(ary.join("+"));
      console.log(ary);


      //扩展运算符 ...
      //1.将数组变成非数组  将数组的中括号去掉展开数组
      //2.将非数组(只要是有length 有索引的)变成数组
      let arr1=[1,2,3];
      let arr2=[10,20,30];
      console.log(arr1.concat(arr2));
      console.log([...arr1, ...arr2]);

      let arr3=[1,13,45,63,80,57];
      //求数组中最大的值
      Math.max(1,13,45,63,80,57);
      console.log(Math.max(...arr3));

      //["js",1,2,3,"css"]
      ["js",...arr1,"css"]


      //2.字符串,参数集合arguments,元素集合
      let s1="12345";
      console.log([...s1]);
      function f() {
        console.log([...arguments]);
      }
      f(1,2,3,4,5)
  </script>
```

2.数组的解构赋值

```
<script>
    let ary1=[1,2,3];
    let x1,x2,x3;
    // x1=ary1[0];
    // x2=ary1[1];
    // x3=ary1[2];
    [x1,x2,x3]=[1,2];
    console.log(x1);
    console.log(x2);
    console.log(x3);

    //[变量,变量]=数组

    let [y1,y2,y3]=["JS","CSS"];

    var a=12;
    var b=13;

    // var c=a;//c=12;
    // a=b;//a=13
    // b=c;//b=12

    // a=a+b;//12+13=25
    // b=a-b;//25-13=12
    // a=a-b;//25-12=13

    //交换变量的值
    [a,b]=[b,a];

    //省略赋值
    let [,,n,m]=[1,2,3,4];

    //不定参数赋值
    let [n1,...n2]=[0,12,13,14,15,16];
    //n1=0;
    //n2=[12,13,14,15,16]
</script>
```

3.数组去重

```
<script>
    let ary=[1,22,13,14,1,2,12,13,14,1,1,0,5,3,2];
    //循环原数组获取数组中的每一项(for,for of,forEach,map),将每一项往新数组
中放(push) 但是 放之前需要判断一下(includes),只有新数组中没有这一项才放进去
    /*
    let arr=[];
    for (let i=0;i<ary.length;i++){
      if (!arr.includes(ary[i])){
        arr.push(ary[i]);
      }
    }
    console.log(arr);


    let arr=[];
    for (let item of ary){
      if(!arr.includes(item)){
        arr.push(item)
      }
    }


    let arr=[];
    ary.forEach(function (item) {
      if(!arr.includes(item)){
        arr.push(item)
      }
    });
    console.log(arr);



    let arr=ary.filter(function (item, index) {
      //判断在index之前的项组成的数组(ary.slice(0,index))中有没有item
      //有 不留下,没有留下
      return !ary.slice(0,index).includes(item)
    });
    console.log(arr);
    */

    for (let i=0;i<ary.length;i++){
      if(i!=ary.lastIndexOf(ary[i])){
        ary.splice(i,1);
        i--;
      }
    }
    console.log(ary);
</script>
```

4.数组去重（利用对象）

```
<script>
```

```
    //利用对象的属性名不可以重复的特点,让对象的属性名和属性值都变成数组的每一项
    let ary=[1,1,2,1,1,3,4,2,1,4,3];
    let obj={};
    for (let item of ary){
      obj[item]=item;
    }
    let arr=[];
    for (let key in obj){
      //将属性值放进数组中 key属性名是字符串
      arr.push(obj[key])
    }
    console.log(arr);
</script>
```

5.遍历接口

```
<script>
    let ary=["js","css","vue","react","node"];
    for (let item of ary) {
      //item是数组的每一项
    }

    //ary.keys() 提供你去遍历数组索引的接口
    for (let index of ary.keys()) {
      console.log(index);
    }

    //ary.entries() 提供你去遍历数组索引和每一项的接口

    for (let e of ary.entries()){
      //e 是一个数组 [索引,当前项]
      console.log(e);
    }
    //分别拿到索引和每一项
    for (let [index,item] of ary.entries()){
      //利用数组的解构赋值 分别获取
      //index:索引
      //item :当前项
      //[index,item]=[0,"js"]
      //[index,item]=[1,"css"]
      //[index,item]=[2,"vue"]
      //.....
      console.log(index,item);
    }
</script>
```

6.冒泡排序

```
<script>
  let ary = [1, 34, 16, 3, 18, 10];
  for (var i = 1; i < ary.length; i++) {
    for (var j = 0; j < ary.length - i; j++) {
      if (ary[j] > ary[j + 1]) {
        [ary[j], ary[j + 1]] = [ary[j + 1], ary[j]]
      }
    }
  }
  ;
  console.log(ary);
</script>
<script>

  arr = [100, 34, 16, 3, 18, 70];
  //第一轮 i=1;
  //j=0    arr[j]>arr[j+1]  arr[0]>arr[1]   100>34   交换位置  [34,10
0,16,3,18,70];
  //j=1    arr[j]>arr[j+1]  arr[1]>arr[2]   100>16   交换位置  [34,1
6,100,3,18,70];
  //j=2    arr[j]>arr[j+1]  arr[2]>arr[3]   100>3    交换位置  [34,1
6,3,100,18,70];
  //j=3    arr[j]>arr[j+1]  arr[3]>arr[4]   100>18   交换位置  [34,1
6,3,18,100,70];
  //j=4    arr[j]>arr[j+1]  arr[4]>arr[5]   100>70   交换位置  [34,1
6,3,18,70,100,];
  //j<5  arr.length-i

  //第二轮 i=2;  [34,16,3,18,70,100,];
  //j=0    arr[j]>arr[j+1]  arr[0]>arr[1]   34>16    交换位置   [16,3
4,3,18,70,100,];
  //j=1    arr[j]>arr[j+1]  arr[1]>arr[2]   34>3     交换位置   [1
6,3,34,18,70,100,];
  //j=2    arr[j]>arr[j+1]  arr[2]>arr[3]   34>18    交换位置   [1
6,3,18,34,70,100,];
  //j=3    arr[j]>arr[j+1]  arr[3]>arr[4]   34<70    不交换位置 [1
6,3,18,34,70,100,];
  //j<4

  //第三轮 i=3;  [16,3,18,34,70,100,];
  //j=0    arr[j]>arr[j+1]  arr[0]>arr[1]   16>3     交换位置   [3,1
6,18,34,70,100,];
  //j=1    arr[j]>arr[j+1]  arr[1]>arr[2]   16<18    不交换位置 [3,1
6,18,34,70,100,];
  //j=2    arr[j]>arr[j+1]  arr[2]>arr[3]   18<34    不交换位置 [3,1
6,18,34,70,100,];

  let f=null;
  for (var i = 1; i < arr.length; i++) {
    f=true;
    for (var j = 0; j < arr.length - i; j++) {
      if (arr[j] > arr[j + 1]) {
        f=false;
        [arr[j], arr[j + 1]] = [arr[j + 1], arr[j]];
      }
    }
    if(f)break;
  }
  console.log(arr);

</script>
```

7.插入排序

```
<script>
    let ary=[3,12,5,7,8,34,1,10,16,57,32];//桌上的牌
    let ary1=[];//手里的牌
    //先摸一张牌放在数组中
    ary1[0]=ary[0];//手里的牌是 [3]
    for (var i=1;i<ary.length;i++){
      //摸到的牌是 ary[i]
      //将摸到的牌ary[i] 去跟手里的牌比较 从后往前比较
      for(var j=ary1.length-1;j>=0;j--){
        //一旦发现摸到的牌ary[i] 比手里某一张牌大了

        if(ary[i]>ary1[j]){
          //将摸到的牌ary[i]插入到 ary1[j]的后面
          ary1.splice(j+1,0,ary[i]);
          //一旦插入后面就不需要比较了
          break;
        }
        //当手里的牌一直比较到索引为0的时候还没有发现比他小的
        if(j==0){
          //将摸到的牌ary[i]放在数组ary1的最前面
          ary1.unshift(ary[i]);
        }
      }
    }
    console.log(ary1);
</script>
```