

数组 (重点)、循环嵌套





今日单词

模块	说明	单词	作用
数组	数组长度	数组.length	得到数组的长度
	新增数组元素	数组.push(新增元素)	数组末尾新增元素
		数组.unshift(新增元素)	数组开头新增元素
	删除数组元素	数组.pop()	删除数组最后一个元素
		数组.shift()	删除第一个数组元素
	新增/删除数组元素	数组.splice()	从指定位置删除/新增数组元素
	数组排序	数组.sort()	对数组进行升序/降序排序





- ◆ 数组
- ◆ 循环嵌套
- ◆ 综合案例





数组

- 数组的基本使用
- 操作数组
- 数组案例



数组(Array) - 重点



作用: 在单个变量名下存储多个数据

例如我要保存咱们班同学的姓名:

let name1 = '小明'

let name2 = '小刚'

let name3 = '小红'

let name4 = '小丽'

let name5 = '小米'

let names = ['小明', '小刚', '小红', '小丽', '小米']





数组的基本使用

1. 声明语法

let 数组名 = [数据1, 数据2, ..., 数据n]

let names = ['小明', '小刚', '小红', '小丽', '小米']

- 注意事项:
- » 数组是按顺序保存(<mark>有序</mark>),所以每个数据都有自己的编号
- » 编号从 0 开始,数据的编号经常称为 **索引或下标**
- 数组可以存储任意类型的数据



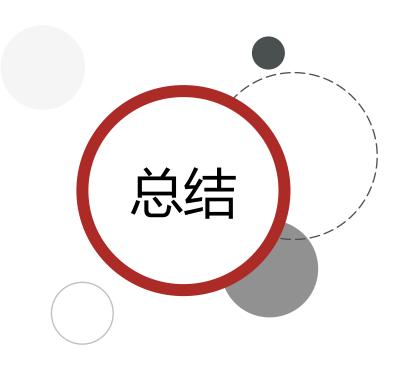
数组的基本使用

2. 取值语法

数组名[下标]

```
let names = ['小明', '小刚', '小红', '小丽', '小米']
names[0] // 小明
names[1] // 小刚
```





- 1. 数组的作用是什么?
 - > 在单个变量名下存储多个数据
 - ▶ 数组也是一种数据类型,属于引用数据类型
- 2. 数组如何声明的?

```
let names = ['小明', '小刚', '小红', '小丽', '小米']
```

- ▶ 数组有序的,索引号从 0 开始
- > 数组里面可以存放任意数据类型
- 3. 数组如何取值?
 - ▶ 数组名[下标]

```
let names = ['小明', '小刚', '小红', '小丽', '小米']
names[0] // 小明
names[1] // 小刚
```



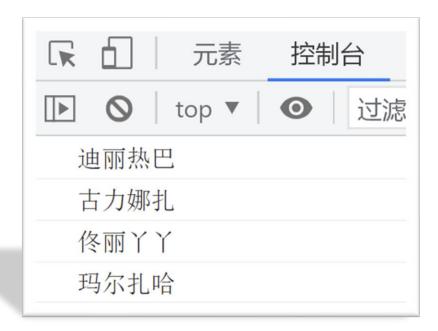
1 练习

• 数组练习

需求:

①:请声明一个数组,里面包含4个数据: '迪丽热巴','古力娜扎','佟丽丫丫','玛尔扎哈'

②:请依次将4个数据控制台打印输出





数组的基本使用

3. 遍历数组 (重点)

■ 遍历数组: 把数组中每个数据都访问到

● 使用场景:

```
let arr = ['迪丽热巴', '古力娜扎', '佟丽丫丫', '玛尔扎哈']
console.log(arr[0])
console.log(arr[1])
console.log(arr[2])
console.log(arr[3])
```

● 我们可以使用for循环来遍历数组,其实遍历数组也是for循环重要的使用场景





数组的基本使用

3. 遍历数组 (重点)

数组长度

数组中数据的个数,通过数组.length属性获得

数组元素

数组中数据,也称为数组元素

```
let arr = ['迪丽热巴', '古力娜扎', '佟丽丫丫', '玛尔扎哈']
for (let i = 0; i < arr.length; i++) {
    console.log(arr[i])
}
```



数组求和

需求: 求数组 (arr) [2, 6, 1, 7, 4] 里面所有元素的和以及平均值

核心:

arr[**0**]

arr[1] arr[2] arr[3]

arr[4]



軍 案例

数组求和

需求:求数组[2,6,1,7,4]里面所有元素的和以及平均值

分析:

①:声明一个求和变量 sum,一个平均值变量 average

②: 遍历这个数组, 把里面每个数组元素加到 sum 里面

③:用求和变量 sum 除以数组的长度就可以得到数组的平均值 average



数组求最大值

需求: 求数组 [2, 6, 1, 25, 7] 中的最大值

分析:

①:声明一个保存最大元素的变量 max, 默认最大值可以取数组中的第一个元素

②:遍历这个数组,把里面每个数组元素和 max 相比较,把比较的最大的放入max变量里面

③: 最后输出这个 max 就是最大值

拓展:

自己求一下最小值





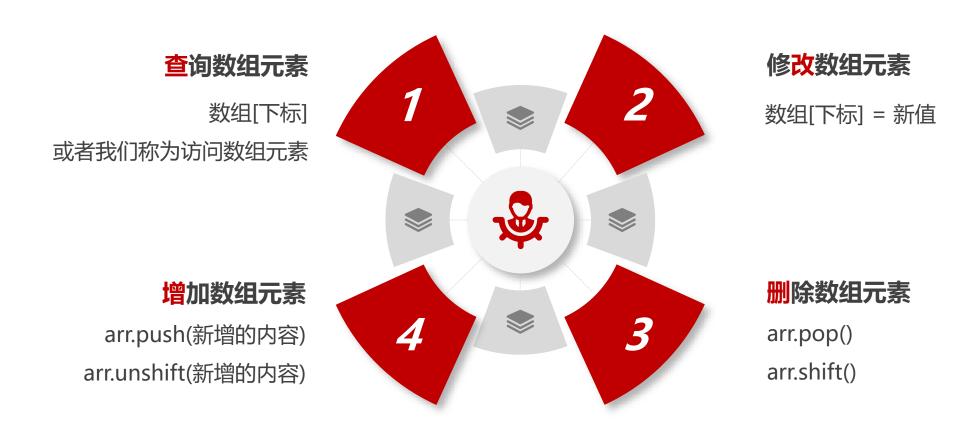
数组

- 数组的基本使用
- 操作数组
- 数组案例



操作数组

数组本质是数据集合,操作数据无非就是增删改查语法:





操作数组 - 查和改

- 查询语法:
- ▶ 数组[下标]
- > 返回值:如果查询不到则返回 undefined

● 修改语法:

- ▶ 数组[下标] = 新值
- ▶ 返回值:如果下标不存在,则是新增一个数组元素,并修改了数组长度(尽量避免)



2.2 操作数组 - 新增元素

数组.push(新增数据)

作用:将一个或多个元素添加到数组的末尾 (重点)

返回值:该数组的新长度,会修改原数组

新增

```
let arr = ['red', 'green']
arr.push('pink')
console.log(arr) // ['red', 'green', 'pink']
```

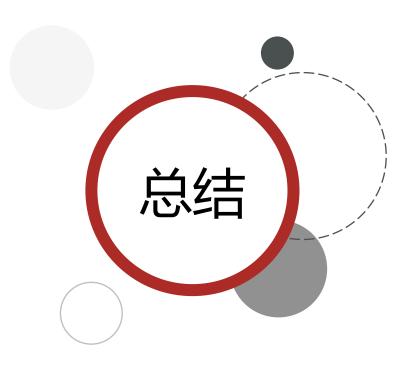
数组.unshift(新增数据)

作用:将一个或多个元素添加到数组的开头 **返回值**:该数组的新长度,会修改原数组

```
let arr = ['red', 'green']
arr.unshift('pink')
console.log(arr) // ['pink', 'red', 'green']
```







- 1. 想要数组末尾增加数据元素利用哪个方法?
 - > arr.push(新增内容)
 - > 可以添加一个或者多个数组元素
 - > 返回的是数组新长度
- 2. 想要数组开头增加数据元素利用哪个方法?
 - > arr.unshift(新增内容)
 - > 可以添加一个或者多个数组元素
 - > 返回的是数组长度
- 3. 两种方法会修改原数组吗?
 - > 会的



富案例

数组筛选

需求: 将数组 [2, 0, 6, 1, 77, 0, 52, 0, 25, 7] 中大于等于 10 的元素选出来, 放入新数组中

分析:

①:声明一个新的数组用于存放新数据 newArr

②:遍历原来的旧数组, 找出大于等于 10 的元素,依次追加push给新数组 newArr



数组去0案例

需求: 将数组 [2, 0, 6, 1, 77, 0, 52, 0, 25, 7] 筛选, 形成一个不包含 0 的新数组

分析:

①:声明一个新的数组用于存放新数据 newArr

②:遍历原来的旧数组, 找出不等于0的元素,依次追加push给新数组 newArr



操作数组 - 删除元素

删除

数组.pop()

作用: 从数组中删除最后一个元素,修改原数组

返回值:该元素的值

let arr = ['red', 'green']
arr.pop()
console.log(arr) // ['red']

数组.shift()

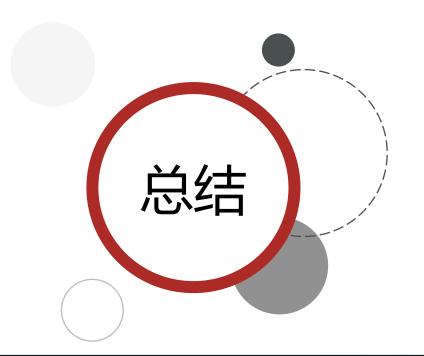
作用: 从数组中删除第一个元素, 修改原数组

返回值:该元素的值

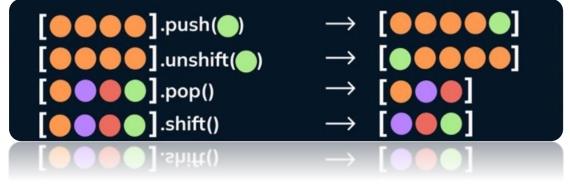
```
let arr = ['red', 'green']
arr.shift()
console.log(arr) // ['green']
```







- 1. 想要数组末尾删除一个数据元素利用哪个方法?
 - > arr.pop()
 - > 返回的是该数组元素
- 2. 想要数组开头删除一个数据元素利用哪个方法?
 - arr.shift()
 - > 返回的是该数组元素
- 3. 两种方法会修改原数组吗?
 - > 会的





操作数组

数组. splice() 方法 删除数组元素

说明: 可以指定位置删除数组元素,注意它修改原数组

splice(start, deleteCount)

- > start 起始位置:
 - ▶ 指定修改的开始位置(从0计数)
- > deleteCount 删除个数:
 - 表示要移除的数组元素的个数
 - ▶ 可选的。 如果省略则默认从指定的起始位置删除到最后





手风琴案例

需求: 利用循环生成图片





手风琴案例

```
需求: 利用循环生成图片
数组数据:
let arr = [
       './images/1.jpg',
       './images/2.jpg',
       './images/3.jpg',
       './images/4.jpg',
       './images/5.jpg',
       './images/6.jpg',
       './images/7.jpg',
```



手风琴案例

小技巧: 利用循环拼接字符串 (原理跟累加和类似)

①: 声明一个空的字符串 str = "

②:循环里面利用 += 进行拼接

③:把拼接完毕的字符串放入容器中

<div></div>



手风琴案例

小技巧: 利用循环拼接字符串 (原理跟累加和类似)

<div></div>
<div></div>

①: 声明一个空的字符串 str = "

②:循环里面利用 += 进行拼接

③:把拼接完毕的字符串放入容器中



手风琴案例

小技巧: 利用循环拼接字符串 (原理跟累加和类似)

①:声明一个空的字符串

②:循环里面利用 += 进行拼接

③:把拼接完毕的字符串放入容器中

```
<div><img src="./images/1.jpg" alt=""></div>
<div><img src="./images/2.jpg" alt=""></div>
<div><img src="./images/3.jpg" alt=""></div>
<div><img src="./images/4.jpg" alt=""></div>
<div><img src="./images/4.jpg" alt=""></div>
<div><img src="./images/5.jpg" alt=""></div>
<div><img src="./images/6.jpg" alt=""></div>
<div><img src="./images/6.jpg" alt=""></div>
<div><img src="./images/7.jpg" alt=""></div>
</div>
```

<div class="box">

</div>





- ◆ 数组
- ◆ 循环嵌套
- ◆ 综合案例







循环嵌套

循环嵌套: 一个循环语句里面又包含另一个循环语句

// 要重复执行的代码



1. 循环嵌套

循环嵌套: 一个循环语句里面又包含另一个循环语句

```
for (初始化; 循环条件; 变量计数) {
// 要重复执行的代码
}
```



循环嵌套

循环嵌套: 一个循环语句里面又包含另一个循环语句

```
for (初始化; 循环条件;变量计数) {
    for (初始化; 循环条件; 变量计数) {
        // 要重复执行的代码
    }
}
```

假如每天记住5个单词,连续记3天 代码如何实现?

第1天 记住第1个单词 记住第2个单词 记住第3个单词 记住第4个单词 记住第5个单词 第2天 记住第1个单词 记住第2个单词 记住第3个单词 记住第4个单词 记住第5个单词 第3天 记住第1个单词 记住第2个单词 记住第3个单词 记住第4个单词 记住第5个单词



循环嵌套

● 循环嵌套: 一个循环语句里面又包含另一个循环语句,一般出现于for循环中

```
for (初始化; 循环条件;变量计数) {
    for (初始化; 循环条件; 变量计数) {
        // 码要重复执行的代码
    }
}
```

- 嵌套循环特点:
- 外部循环每循环一次,内部循环执行所有次





生成电影院座位

需求: 根据 seats 数组, 生成电影院座位

```
let seats = [
    [0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0],
    [0, 0, 1, 1, 1, 1, 0, 0],
    [0, 0, 1, 1, 0, 1, 1, 0, 0],
    [0, 1, 1, 1, 1, 0],
    [0, 0, 0, 0, 0, 0],
]
```

外层for循环表示行 内层for循环表示列 (座位)







- ◆ 循环-for
- ◆ 数组
- ◆ 综合案例





根据数据生成快捷导航

需求:用户输入2个导航的数据,结合原本快捷导航数据,一并渲染到页面上





根据数据生成快捷导航

需求:用户输入2个导航的数据,结合原本快捷导航数据,一并渲染到页面上

分析:

①: 声明一个数组, 里面放原本的快捷导航数据

②:需要输入2次,将2次用户输入的数据添加到原本的数组中

③:遍历该数组,根据数据生成快捷导航,渲染打印到页面中

▶ 声明一个空字符串 str

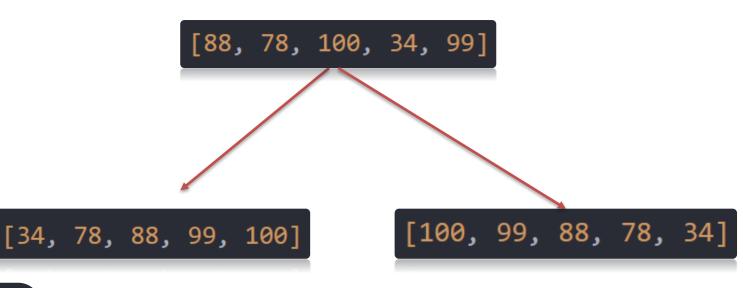
▶ 根据数组利用循环拼接字符串,生成多个快捷导航

> 把生成的快捷导航添加到 nav 盒子中



拓展 - 数组排序

- 数组. sort() 方法可以排序
- 语法:
- 数组.sort() 会修改原数组



let arr = [4, 2, 5, 1, 3]
// 1.升序排列写法
arr.sort(function (a, b) {
 return a - b
})
console.log(arr) // [1, 2, 3, 4, 5]
// 降序排列写法
arr.sort(function (a, b) {
 return b - a
})
console.log(arr) // [5, 4, 3, 2, 1]

升序排序

降序排序



今日单词

模块	说明	单词	作用
数组	数组长度	数组.length	得到数组的长度
	新增数组元素	数组.push(新增元素)	数组末尾新增元素
		数组.unshift(新增元素)	数组开头新增元素
	删除数组元素	数组.pop()	删除数组最后一个元素
		数组.shift()	删除第一个数组元素
	新增/删除数组元素	数组.splice()	从指定位置删除/新增数组元素
	数组排序	数组.sort()	对数组进行升序/降序排序



传智教育旗下高端IT教育品牌