# JavaScript 基础第三天

数组







- 1. 掌握循环语句,让程序具备重复执行能力
- 2. 掌握数组声明及访问的语法





- ◆ 循环-for
- ◆ 数组
- ◆ 综合案例





## 循环-for

- for循环基本使用
- 循环嵌套



#### 1.1 for 循环-基本使用

目标:掌握for循环重复执行某些代码

#### 1. for循环语法

● 作用: 重复执行代码

● **好处:** 把声明起始值、循环条件、变化值写到一起,让人一目了然,它是最常使用的循环形式





- 循环练习
- 1. 利用for循环输出1~100岁
- 2. 求1-100之间所有的偶数和
- 3. 页面中打印5个小星星 ★★★★
- 4. for循环的最大价值: 循环数组

需求: 请将数组 ['马超','赵云','张飞','关羽','黄忠'] 依次打印出来



#### 1.1 for 循环-基本使用

目标:掌握for循环重复执行某些代码

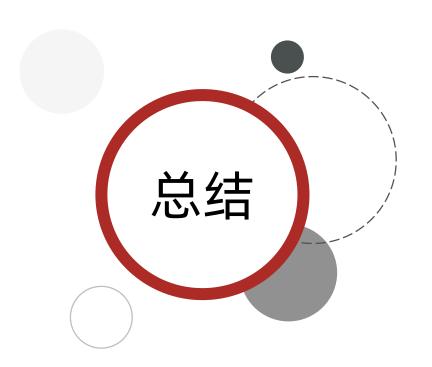
#### 2. 退出循环

- continue 退出本次循环,一般用于排除或者跳过某一个选项的时候,可以使用continue
- break 退出整个for循环,一般用于结果已经得到,后续的循环不需要的时候可以使用

#### 了解:

- 1. while(true)来构造"无限"循环,需要使用break退出循环。
- 2. for(;;) 也可以来构造"无限"循环,同样需要使用break退出循环。





- 1. for循环和while循环有什么区别呢:
  - ➤ 当如果明确了循环的次数的时候推荐使用for循环
  - ➤ 当不明确循环的次数的时候推荐使用while循环





## 循环-for

- · for循环基本使用
- 循环嵌套



#### 1.3 for 循环嵌套

● for 循环嵌套

```
for (外部声明记录循环次数的变量;循环条件;变化值) {
  for (内部声明记录循环次数的变量;循环条件;变化值) {
     循环体
```

▶ 一个循环里再套一个循环,一般用在for循环里





计算: 假如每天记住5个单词,3天后一共能记住多少单词?

拆解:

▶ 第一天: 5个单词

▶ 第二天: 5个单词

▶ 第三天: 5个单词

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





#### • 打印5行5列的星星

需求: 页面中打印出5行5列的星星

分析:

①:利用双重for循环来做

②:外层循环控制打印行,内层循环控制每行打印几个(列)

#### 升级版本:

用户输入行数和列数,打印对应的星星!







### • 打印倒三角形星星

需求: 如图所示

分析:

①:利用双重for循环来做

②:外层循环控制打印行,内层循环控制每行打印几个(列)

③:内层循环的个数跟第几行是一一对应的







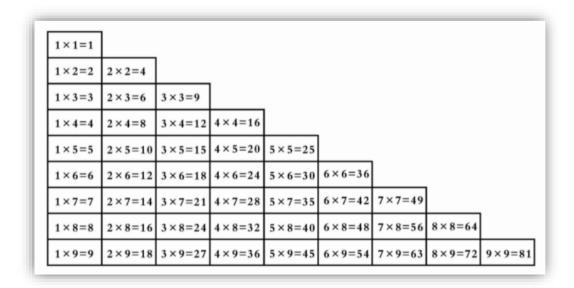
#### • 九九乘法表

需求: 如图所示

分析:

①: 只需要把刚才倒三角形星星做改动

②: ★ 换成 1 x 1 = 2 格式







- ◆ 循环-for
- ◆ 数组
- ◆ 综合案例





## 数组

- 数组是什么
- 数组的基本使用
- 操作数组
- 数组案例



#### 2.1 数组是什么

目标: 能说出数组是什么

● 数组: (Array)是一种可以按顺序保存数据的数据类型

● 为什么要数组?

▶ 思考:如果我想保存一个班里所有同学的姓名怎么办?

▶ 场景: 如果有多个数据可以用数组保存起来, 然后放到一个变量中, 管理非常方便





## 数组

- 数组是什么
- 数组的基本使用
- 操作数组
- 数组案例



目标: 能够声明数组并且能够获取里面的数据

● 1. 声明语法

let 数组名 = [数据1,数据2,...,数据n]

let arr = new Array(数据1,数据2,..数据n)

● 例

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

- 数组是按顺序保存,所以每个数据都有自己的编号
- 计算机中的编号从0开始,所以小明的编号为0,小刚编号为1,以此类推
- 在数组中,数据的编号也叫**索引或下标**
- 数组可以存储任意类型的数据



● 2. 取值语法

### 数组名[下标]

● 例

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

- 通过下标取数据
- 取出来是什么类型的,就根据这种类型特点来访问



#### 3. 一些术语:

▶ 元素:数组中保存的每个数据都叫数组元素

▶ 下标:数组中数据的编号

➤ 长度:数组中数据的个数,通过数组的length属性获得

```
let names = ['小明', '小刚', '小红', '小丽', '小米']
console.log(names[0]) // 小明
console.log(names[1]) // 小刚
console.log(names.length) // 5
```



#### 4. 遍历数组(重点)

目标: 能够遍历输出数组里面的元素

- 用循环把数组中每个元素都访问到,一般会用for循环遍历
- 语法:

```
for (let i = 0; i < 数组名.length; i++) {
数组名[i]
}
```

● 仮

```
let nums = [10, 20, 30, 40, 50]
for (let i = 0; i < nums.length; i++) {
    document.write(nums[i])
}</pre>
```





### • 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。

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。



## 1 练习

#### • 数组求最大值和最小值

需求: 求数组 [2,6,1,77,52,25,7] 中的最大值

分析:

①:声明一个保存最大元素的变量 max。

②: 默认最大值可以取数组中的第一个元素。

③: 遍历这个数组, 把里面每个数组元素和 max 相比较。

④:如果这个数组元素大于max 就把这个数组元素存到 max 里面,否则继续下一轮比较。

⑤: 最后输出这个 max

#### 拓展:

自己求一下最小值





## 数组

- 数组是什么
- 数组的基本使用
- 操作数组
- 数组案例



#### 2.3 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:

### 查询数组数据

数组[下标]

或者我们称为访问数组数据





### 重新赋值

数组[下标] = 新值

### 数组添加新的数据

arr.push(新增的内容)
arr.unshift(新增的内容)





#### 删除数组中数据

arr.pop()

arr.shift()

arr.splice(操作的下标,删除的个数)



#### 2.3 操作数组-新增

目标:掌握利用push向数组添加元素(数据)

- 数组.push() 方法将一个或多个元素添加到数组的末尾,并返回该数组的新长度 (重点)
- 语法:

```
arr.push(元素1, ..., 元素n)
```

● 例如:

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

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



#### 2.3 操作数组-新增

- arr.unshift(新增的内容) 方法将一个或多个元素添加到数组的开头,并返回该数组的新长度
- 语法:

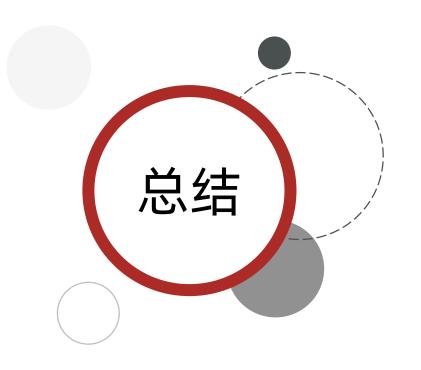
```
arr.unshift(元素1, ..., 元素n)
```

● 例如:

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

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





- 1. 想要数组末尾增加数据元素利用那个方法?
  - arr.push()
  - ▶ 可以添加一个或者多个数组元素
  - > 返回的是数组长度
- 2. 想要数组开头增加数据元素利用那个方法?
  - arr.unshift()
  - > 可以添加一个或者多个数组元素
  - > 返回的是数组长度
- 3. 重点记住那个?
  - arr.push()



## 国 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

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

③:依次追加给新数组 newArr



## 国 案例

### 数组去0案例

需求: 将数组 [2, 0, 6, 1, 77, 0, 52, 0, 25, 7] 中的 0 去掉后, 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

③:依次追加给新数组 newArr



#### 2.3 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:

### 查询数组数据

数组[下标]

或者我们称为访问数组数据





### 重新赋值

数组[下标] = 新值

### 数组添加新的数据

arr.push(新增的内容)
arr.unshift(新增的内容)





#### 删除数组中数据

arr.pop()

arr.shift()

arr.splice(操作的下标,删除的个数)



目标: 能够删除数组元素(数据)

- 数组. pop() 方法从数组中删除最后一个元素,并返回该元素的值
- 语法:

```
arr.pop()
```

● 例如:

```
let arr = ['red', 'green']
arr.pop()
console.log(arr) // ['red']
couzote:tog(sul) \\ [,usq.]
```



- 数组. shift() 方法从数组中删除第一个元素,并返回该元素的值
- 语法:

```
arr.shift()
```

● 例如:

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



#### ● 需求使用场景:

- 1. 随机抽奖,中奖的用户就需要从数组里面删除,不允许重复抽奖
- 2. 点击删除按钮,相关的数据会从商品数据中删除
- 后期课程我们会用到删除操作,特别是 splice

	随机问答	<b>F</b>
问题是: 赵云		
	开始	结束人





- 数组. splice() 方法 删除指定元素
- 语法:

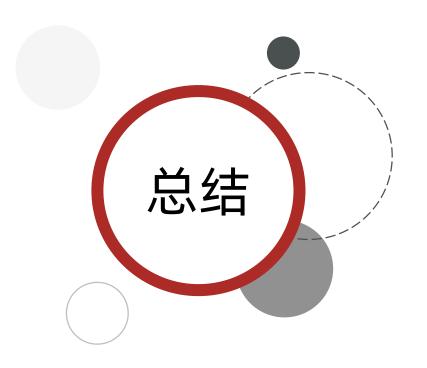
arr.splice(start, deleteCount)

arr.splice(起始位置,删除几个元素)

#### 解释:

- ➤ start 起始位置:
  - ▶ 指定修改的开始位置(从0计数)
- deleteCount:
  - ▶ 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





- 1. 想要数组末尾删除1个数据元素利用那个方法? 带参数吗?
  - arr.pop()
  - > 不带参数
  - > 返回值是删除的元素
- 2. 想要数组开头删除1个数据元素利用那个方法? 带参数吗?
  - arr.shift()
  - ▶ 不带参数
  - ▶ 返回值是删除的元素
- 3. 想要指定删除数组元素用那个? 开发常用吗? 有那些使用场景?
  - ➤ arr.splice(起始位置, 删除的个数)
  - ▶ 开发很常用,比如随机抽奖,比如删除指定商品等等。





- ◆ 循环-for
- ◆ 数组
- ◆ 综合案例





### 根据数据生成柱形图

需求: 用户输入四个季度的数据,可以生成柱形图





## 国 案例

#### 根据数据生成柱形图

需求: 用户输入四个季度的数据,可以生成柱形图

分析:

①:需要输入4次,所以可以把4个数据放到一个数组里面

利用循环,弹出4次框,同时存到数组里面

②:遍历改数组,根据数据生成4个柱形图,渲染打印到页面中

▶ 柱形图就是div盒子,设置宽度固定,高度是用户输入的数据

▶ div里面包含显示的数字和 第n季度



#### 冒泡排序

- 冒泡排序是一种简单的排序算法。
- 它重复地走访过要排序的数列,一次比较两个元素,如果他们的顺序错误就把他们交换过来。走访数列的工作是重复地进行直到没有再需要交换,也就是说该数列已经排序完成。
- 这个算法的名字由来是因为越小的元素会经由交换慢慢"浮"到数列的顶端。
- 比如数组 [2,3,1,4,5] 经过排序成为了 [1,2,3,4,5] 或者 [5,4,3,2,1]



## 案例

### 冒泡排序

分析:

5 4 3 2 1

1. 一共需要的趟数 我们用外层for 循环

5个数据我们一共需要走4趟 长度就是数组长度减去1 arr.length - 1

一趟交换次数 我们 用里层 for循环

第一趟交换 4次

第二趙交换 3次

第三趙交換 2次

第四趙 交换 1次

长度就是 数组长度 减去 次数

但是我们次数是从 0次开始的 所以 最终 arr.length - i - 1

3. 交换2个变量就好了





4 3 2 1 5

第2趟:

第1趟:

3 2 1 4 5

第3趟:

2 1 3 4 5

第4趟:

1 2 3 4 5



#### 数组排序

- 数组. sort() 方法可以排序
- 语法:

```
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]
```

```
// 1.升序排列写法
return a - b
console.log(arr) // [1, 2, 3, 4, 5]
// 降序排列写法
console.log(arr) // [5, 4, 3, 2, 1]
```





- 1. 晚自习回来每个同学先必须xmind梳理今日知识点 (md 笔记也行)
- 2. 需要把今天的所有案例,按照书写顺序写一遍。综合案例写三遍。
- 3. 开始做测试题: 扫码 ‖ PC端: https://ks.wjx.top/vj/YDMWN1o.aspx
- 4. 独立书写今日作业, 见附件: 06-作业
- 5. 每日一句鼓励自己的话







传智教育旗下高端IT教育品牌