正课:

1. \*\*\*循环

2. \*\*\*数组

1. \*\*\*循环:

什么是: 让程序反复执行相同代码段

何时: 只要让程序反复执行一项任务时

如何: 三要素:

1. 循环条件: 让循环可以继续执行的条件

只要循环条件满足，就可以反复执行循环

直到循环条件不满足时，就退出循环

2. 循环变量: 循环中用作比较和判断的变量

考虑: 1. 从几开始

2. 到几结束

3. 每次递增/递减几

3. 循环体: 循环要反复执行的代码段

如果循环条件始终为true，则循环永远无法退出——死循环

while循环: 先判断循环条件，再决定是否执行循环

何时: 要求必须先满足条件，才能执行循环时

//声明并初始化循环变量

var 变量=值;

//当满足(循环条件)时{

while(循环条件){

反复执行循环体;

修改循环变量的值;

}

do...while循环: 先至少执行一次循环，再判断循环条件，决定是否反复执行

何时: 只要希望即使条件不满足，也至少能执行一次时

//声明并初始化循环变量

var 变量=值;

do{

反复执行循环体;

修改循环变量的值;

}while(循环条件) ;//当满足(循环条件)时{

while vs do while:

只要第一次条件都满足，则两者完全一样

如果第一次条件不满足，while是一次都不执行

do while至少可执行一次

for循环: 其实就是while循环的简化

何时: 只要循环变量的变化有规律

for(var 变量=值;循环条件;修改循环变量的值){

反复执行循环体;

}

简写: 1. 分支/循环结构中，如果条件之下只有一条语句，则可省略{}, 但强烈不建议这样做——容易出歧义

2. 其实,第一部分可同时声明并初始化多个变量,多个变量间用逗号分隔:

3. 其实,第三部分可同时执行多个短小的语句，用逗号分隔，但是，简写不能破坏原执行顺序

如果省略循环体，则for循环后必须加";"

强调: js中没有块级作用域:

分支结构/循环结构内声明的变量，出了结构依然可用！

循环嵌套:

什么是: 一个循环的循环体内，又执行了另一个内层循环

2. \*\*\*数组Array:

什么是: 内存中连续存储多个数据的存储空间，再起一个名字

vs 变量: 只能存一个值

何时: 只要保存多个相关的数据时，必须用数组

为什么: 为了便于查找和维护数据

程序=数据结构+算法

数据结构: 数据在内存中的存储结构

算法: 解决问题的步骤

好的数据结构可极大提高程序的执行效率

如何:

创建: 3种:

1. 创建空数组:

何时: 如果创建数组时，暂时不知道数组中的内容

如何: var arr=[];

var arr=new Array();

2. 创建数组同时初始化数组的内容

何时: 如果创建数组时已经知道数组的内容

如何: var arr=[值1,值2,...];

var arr=new Array(值1,值2,...)

3. 创建n个空元素的数组:

何时: 如果创建数组时，只知道数据的个数，暂时不知道数据的内容。

如何: var arr=new Array(n);

访问:

概念: 元素: 数组中每个数据，就称为元素

下标index: 每个元素的存储位置的序号

默认都是数字，从0开始，连续不重复

取值: 数组名[下标]

数组中每个元素的用法和普通变量完全一样

数组也称为: 一组变量的集合，起一个统一的变量名。

赋值: 数组名[下标]=值

js中的数组: 三个不限制:

1. 不限制元素的数据类型

2. 不限制下标越界: 不报错!

取值时越界: 仅返回undefined

赋值时越界: 在指定的新位置自动创建新元素保存新值

3. 不限制元素个数: 可随时向任意位置添加新元素

length属性: 记录数组中理论上的元素个数

不一定和实际元素个数相符。

特点: length永远等于最大数字下标+1

自动维护，不需要手动修改

如何: arr.length

强调: length属性属于数组，不能脱离数组单独使用

固定套路:

1. 获得最后一个元素: arr[arr.length-1]

获得倒数第n个元素: arr[arr.length-n]

2. 向末尾追加一个新元素: arr[arr.length]=值

3. 可改小length，达到删除数组末尾元素的目的:

删除数组末尾n个元素: arr.length-=n

... 1个元素: arr.length--

本质: 数组是引用类型的对象!

重新审视按值传递:

什么是: 当变量间赋值时，或将变量作为参数传入函数中时，其实只是将原变量的值赋值一个副本给对方

对于原始类型: 修改新变量，不影响原变量

因为将原变量中的实际值复制一个副本给对方

对于引用类型: 用新变量修改对象，等效于直接修改原对象，也影响原变量。

因为仅将原变量中的地址值复制给新变量，导致两个变量共用同一个对象。

遍历:

什么是: 依次对数组中每个元素执行相同的操作

何时: 只要对数组中每个元素执行相同操作

如何: 循环

for(var i=0;i<arr.length;i++){

arr[i] //当前元素

}

dir问题: 不会立刻输出结果  
 只有手动点击三角时，才动态查找内存中对象的最终结构。会打乱程序的输出结果顺序。

解决: 如果向查看数组中间状态的内容:

console.log(String(arr)); //对数组拍照

总结: 如果查看中间状态: console.log(String(arr));

如果查看结构: console.dir(arr)