**07**

**数组**

**1.数组概述**

程序=数据+算法

所谓的数据结构,就是把数据和数据之间的关系按照特定的结构来保存.

数组就是一中特定形式的数据,是数据的集合

**2.创建数组:两种方法(1),(2)**

**(1)数组字面量**

var names = ['Tom','Jerry'];

**(2)构造函数**

|  |
| --- |
| ①var names = new Array(3);//创建一个数组，有3个数组元素  ②var names = new Array('tom','jerry');//创建数组，然后初始化值 |

**（3）数组元素的访问**

数组中每一个元素默认都有下标，是从0开始

访问数组，就可以通过下标

数组[下标] names [0]

**(4)数组的长度**

length属性可以获取数组的长度，使用length还可以往数组里面添加元素names[names.length] = 'm';

访问数组中的最后一个元素names[names.length-1]

**3.对比null和undefined**

相同点：都是原始类型数据，保存在栈内存中

不同点：

undefined：表示当一个变量声明未赋值；声明函数，没有返回值或者返回值为空；访问对象中不存在的属性；访问数组中不存在的元素；

null：本来一个变量指向引用类型的数据，赋值为null断开所指向的引用类型；称为**释放对象的引用。**

**4.遍历数组**

（1）for in

for(var proNmae in arr){

//proName数组中每个元素的下标

//arr[proName] 数组中的每一个元素

}

数组也是一种对象，可以称为数组对象.

(2)for循环,所遍历的是数组元素的下标;

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

//i数组的下标

//arr[i] 数组的元素

}

**5.数组的分类**

数组分为关联数组和索引数组

索引数组：以数值作为下标

关联数组：以字符串作为下标，只能使用for-in来遍历数组。

练习1：声明函数，在函数中传递参数(数组)，获取传递数组中的最大值并返回.

练习2：声明函数，传递参数为数组，获取最小值

练习3：声明函数，传递两个参数（参数1数组，参数2值），返回值在数组中的下标。

练习4：对数组进行有小到大的排序（90,32,57,49,80）采用冒泡排序

两层for循环，两个元素值的交换

**6.数组对象的方法（api）（Application Programming Interface）**

toString() 将数组转换成字符串

join() 将数组转换成字符串，可以按照指定的格式拼接数组元素

concat() 拼接两个或者更多的数组

slice() 截取数组

splice() 修改数组（包含删除，替换，插入）

sort() 数组排序，默认就是按照Unicode码从小到大

按照数值从小到大

sort(function(a,b){

return a-b; //从小到大

return b-a;//从大到小

});

push() 在数组的最后添加元素

pop() 删除数组的最后一个元素

练习：创建数组，包含若干个员工对象的信息，每一个员工具有编号，姓名，性别，生日，工资，电话，等属性。使用for/in 遍历数组，每个对象成一行显示

**08**

二维数组

所谓的二维数组，一个数组中，嵌套多个数组。

**今日学习目标**

字符串操作（重点）

匹配模式（掌握）

Math 对象(掌握)

**1.字符串操作**

包装对象（类型）：

目的就是让原始数据类型也可以向引用数据类型一样具有属性和方法

js中有三种包装类型：String类型，Number类型，Boolean类型

（1）转义字符

把普通的字符串，转义成具有特定意义的字符（\）

\n 换行

\t 制表符（tab键）

\' 产生有一个单引号 ( ' )

\\ 产生一个普通的 \

(2)字符串常用操作

toUpperCase();英文字符串转成大写

toLowerCase();英文字符串转成小写

charAt(n) 获取下标对应的字符

charCodeAt(n) 获取指定下标字符的Unicode码

indexOf(values,start) 查找某个字符对应的下标，start就是从哪开始，如果找不到返回-1

lastindexOf(values,start)同indexOf一致。区别在于，查找最后一次出现的位置

slice(start,end)截取字符串，从start开始到end结束，不包含end，如果end为空则截取到最后

substring(start,end)截取字符串，从start开始到end结束，不包含end，如果end为空，截取到最后,

如果为负数，自动转换为0

split() 按照指定格式分割字符串，返回一个数组

**练习：**

初始化4个英文字符，弹出提示框输入4个字符，如果输入的不正确，再次弹出提示框，直到输入正确（不区分大小写）。

**2.匹配模式**

作用：用于查找，替换字符

replace(m,n);用于查找并替换某一个字符串，m要查找的字符串，格式为/google/ig, n是要替换的字符串

i ： ignore忽略

g ：global 全局

match(m); 用于查找匹配的字符串，格式 /google/ig,返回一个数组

search(m); 用于查找匹配的字符串，格式/google/i,返回满足条件的第一个下标，如果找不到返回-1

**3.Math 对象**

Math对象不需要使用new来创建，直接使用。

Math.PI 获取圆周率 （属性）

成员方法：

Math.abs() 取绝对值

Math.ceil() 向上取整 3.1--4

Math.floor() 向下取整 3.9--3

Math.round() 四舍五入取整

Math.max() 取多个数字的最大值

Math.min() 取多个数字的最小值

Math.random()获取随机数，范围0~1之间

练习：模拟彩票摇奖，随机在1-30之间取出7个数字

乘以30 向上取整

**复习**

二维(多维)数组

字符串操作:包装对象,具有属性和方法

转义字符 \n换行 \t制表符 \' \\

**09**

**学习目标**

**Date 对象（重点）**

**Nunber 对象**

**Boolean 对象**

**错误处理**

**ES6(重点)**

**1. Date对象**

Date 对象用于对日期和时间进行存储和计算。

（1）创建（实例化）Date对象

new Date（2018,8,1,10:30:30）；//第二个参数 月，范围0~11

new Date（'2018-9-1 10:30:50'）

new Date(); //当前系统时间对象

Date(); //当前系统时间对象，字符串形式

var d5 = new Date(24\*60\*60\*1000);//相对于计算机元年对象

new Date（d5）;//复制一个Date对象

（2）获取时间信息

getFullYear() 获取年份

getMonth() 获取月份 0-11

getDate() 获取日期

getHours() 获取小时

getMinutes() 获取分钟

getSeconds() 获取秒

getMIlliseconds()获取毫米0-1000

getDay() 获取星期 范围0-6 星期日~星期六

getTime() 获取距离计算机元年的毫秒，常用于计算两个时间差

(3)设置时间信息

setFullYear() 设置年份

setMonth() 设置月份

setDate() 设置日期

setHours() 设置小时

setMinutes() 设置分钟

setSeconds() 设置秒钟

setMillinseconds() 设置毫秒

setTime() 设置距离计算机元年的时间

练习：计算合同到期时间，员工入职时间为2012-6-30，合同有效期3年，求合同到期时间

练习：声明一个函数，格式化当前时间格式，传递参数设置格式；

**2.Number对象**

Number.MAX\_VALUE 获取计算机能存储的最大值

Number.MIN\_VALUE 存储的最小值

toFixd(n) 保留小数点后n位

**3.Boolean**

new Boolean() 本质上是把数据转换成布尔型的值

另一种转换为布尔型！！value-> true / false

**4.错误处理**

SytaxError 语法错误（低级错误），使用了中文输入法，缺少括号等

ReferenceError 引用错误，要使用的变量未声明。

TypeError 类型错误，错误的使用了小括号()等。

RangError 范围错误，参数超出范围

URIError URI（路径）错误

EvalError Eval错误

异常处理：

ReferenceError 和TypeError 两种错误出现后，会影响后续代码的执行

try{

//尝试执行的代码，可能出现错误

}catch{

//err捕获的错误信息

//处理错误的方案

}

**5.ES6新特征 js最新版本**

ECMAScript6

（1）块级作用域

使用let关键字声明的变量

块级作用域：for循环中的{}之间，以及while，do-while中{}使用let声明的变量都是局部变量。ES6新增独立的{}，可以作为块级作用域

（2）箭头函数

回调函数中传递的匿名函数可以使用箭头函数的形式

sort((a,b)=>{

return a-b;

});

如果函数体中只有一行代码并且是return

则 sort（(a,b)=>a-b）;

(3)函数中的参数

在ES6中，函数的形参可以添加默认值，如果没有传递实参，就自动使用形参的值。

function fn(a,b,c=0){ } fn( 1,2 );

(4)模板字符串

任何的符号，包括引号，回车，分号都可以出现在模板字符串之间

**`**之间可以写任何代码${变量名称}**`**

模板字符串之间的变量${变量名称}

练习：获取自己生日的Date对象，如果不是周末提前到上一个周六，打印日期多少