# 数组

## 课程大纲：

阶段一：编程思想；

阶段二：对象；

阶段三：数组；

阶段四：字符串；

阶段五：Math对象

阶段六：Date对象

## 阶段一：编程思想

### 1.1-面向过程：

“**面向过程**”(Procedure Oriented)是一种以过程为中心的编程思想。就是分析出解决问题的所有步骤，然后一步一步的去实现。

### 1.2-面向对象：

对象（工具库）

OOP（Object Oriented Programming）:面向对象编程

“**[面向对象](http://baike.baidu.com/view/125370.htm" \t "http://baike.baidu.com/_blank)**”(Object Oriented,简称OO)是一种以事物为中心的编程思想（用对象！）。是现在编程的主流范式（思想）。核心思想，就是将真实世界存在事物的复杂关系，抽取成一个个不同的对象，然后由这些对象之间的分工与协作，完成对真实世界的模拟。

类：抽象的。

对象：客观（真时存在）存在的！

关系：【对象】是 【类】 的一个 【实例】

## 阶段二：对象

### 2.1-什么是对象？

对象，可以理解为一个工具库，工具库里可以存放很多小工具供我们使用。

### 2.2-对象的创建

var 对象名 = {键名:值} ——》 var 对象名 = new Object();

**键名：也**叫属性名，也就是我们的工具名；

**值：**可以是任何类型的数据，比如数字、字符串、函数表达式、数组、对象等等

**点“.”**操作符：通过点操作符可以调用工具库里的工具

注意：点操作符后面的工具名必须在对象容器（工具库里）存在，并且键名不能写在引号里，以及不能是数字！

**中括号 [“键名”或者变量]，**通过中括号的方式，并在中括号里传入键名，并且键名是为字符串类型或数字类型（数字类型会自动转换为字符串类型），调用工具库里的工具

JavaScript原生提供一个Object对象（注意起首的O是大写），所有其他对象都继承自这个对象。

## 阶段三：数组

### 3.1-什么是数组？

数组是一种数据结构，所为的数据结构，就是计算存储数据的一种方式，而数组的数据结构的特性是：线性集合，就是一组有序的数据，所谓有序的数据指的是数据的下标或索引是基于0的一串有序的数字。

而我们JavaScript里的数组不是通常意义上的数组，而是一个特殊的对象，是用一个对象所模拟而实现的一个含有线性集合特性的一个对象数组。

### 3.2-为什么需要数组？

方便管理和操作数据。

### 3.3-语法：

var 变量名=[]; //数组直接量

**数组的长度：**

变量名.length 可以获取键是基于0的索引下标的所对应的元素（数据）的个数。

数组添加数据：

变量名[索引] = 值；

获取数据：

变量名[索引] //数字索引会自动转换为字符串

var 变量名=new Array(); //new 对象的方式 不常用！

### 3.4-数组的遍历

for;

for in;

ECMAScript: 3.0 ES3.0 (稳定的版本) ES4.0（没发布） ES5.0（有兼容性问题） ES6.0 Node平台

回调函数的概念

forEach; es5 兼容性问题

练习：

求数组中所有元素的和；

求数组的最大数和最小数；

求数组的最小数的下标；

数组去重；

### 3.6-引用数据类型和基本数据类型的区别(扩展)

检测传值的区别。

画内存图理解！

### 3.7-数组常用的方法（工具）

push(数据)：添加数据(一个或多个)到数组的尾部， 返回当前数组的长度。

pop()：删除数组最后一个数据，并把删除的数据返回

unshift(数据)：添加数据（一个或多个）到数组的首部，返回当前数组的长度。

shift():删除数组的第一个数据，并把删除的数据返回

数据结构：（扩展！非重点！了解即可！）

用数组所提供的功能模拟

栈（Last In First Out）LIFO：后进入先出 push pop

队列（First In First Out）FIFO：先进后出 push shift

slice(index):截取,返回一个副本，不该变原来的数组

splice(index1,index2):删除 ，改变数组本身

splice(index1,0,数据):增加

reverse();反转

sort(function(a,b){})；排序 重点！

排序算法：冒泡排序、选择排序、快速排序、归并排序、希尔排序

**冒泡排序() （非重点！要求能够写）**

冒泡排序（Bubble Sort），是一种[计算机科学](http://baike.baidu.com/view/92404.htm" \t "http://baike.baidu.com/_blank)领域的较简单的[排序算法](http://baike.baidu.com/view/297739.htm" \t "http://baike.baidu.com/_blank)。

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大的元素会经由交换慢慢“浮”到数列的顶端，故名。

**选择排序（非重点！要求能够写）**

选择排序（Selection sort）是一种简单直观的[排序算法](http://baike.baidu.com/view/297739.htm" \t "http://baike.baidu.com/_blank)。它的工作原理是每一次从待排序的[数据元素](http://baike.baidu.com/view/38785.htm" \t "http://baike.baidu.com/_blank)中选出最小（或最大）的一个元素，存放在序列的起始位置，直到全部待排序的数据元素排完。

## 阶段四：字符串

### 4.1创建字符串

var str = ”ss”; //string

var str = new String(“ss”); //object

构造函数的概念！

临时对象的概念！

valueOf方法返回包装对象（临时对象）实例对应的原始类型的值。

toString方法返回该实例对应的原始类型值的字符串形式

### 4.2 字符串常用属性（静态属性）和方法（动态属性）

length：获取字符串的字符数；

可以像数组一样使用中括号索引的方式读取指定位置的字符

charAt(index):取一个字符串的指定位置的字符

charCodeAt(index):返回一个字符串中指定位置的字符编码

ASCII（American Standard Code for Information Interchange，美国标准信息交换代码）

Unicode 是ASCII码不能满足使用以后的升级版

GBK包含全部中文字符；UTF-8则包含全世界所有国家需要用到的字符。



String.fromCharCode(94) ://编码转换成字符

indexOf("abc") :查找字符串第一次出现的位置 如果没找到 返回-1

lastIndexOf("abc"): 查找字符串最后一次出现的位置 如果没找到 返回-1

search() 正则匹配 (返回出现的位置)

|  |
| --- |
| **var** str="Visit W3School!"  document.write(str.search(/w3school/i)) |

match stringObj.match(rgExp) (将匹配的内容存入数组)

replace替换字符串 stringObj.replace("tmd", "\*");

这里的替换只能执行一次，不能够进行全局匹配，如果需要全局匹配，则应使用正则表达式

|  |
| --- |
| str.replace(/tmd/gi,"\*"); |
| g表示进行全局匹配，i表示匹配的时候忽略大小写 |

concat():将一个值或多个值

* 字符串比较方法
  + ==和===
  + str=new string('abc') 与var str='abc';
* 字符串的截取方法
  + substring(3，5)
* 字符串分割
  + split——根据分隔符、拆分成数组
  + 案例：内容查找替换功能 .//replace

toLowerCase、toUpperCase

案例：给出一段文字，并抽取关键词所出现的次数？

## 阶段五：Math对象

* + Math.round(3.6) //四舍五入
  + Math.random() //返回0-1之间的随机数
  + Math.max(num1, num2) //返回较大的数
  + Math.min(num1, num2) //返回较小的数
  + Math.abs(num) //绝对值
  + Math.ceil(19.3) //20 向上取整
  + Math.floor(11.8) //11 向下取整
  + Math.pow(x,y) //x的y次方 pow(2,3); //8 2\*2\*2
  + Math.sqrt(num) //开平方 //sqrt(9); //

案例：随机点名

## 阶段六：Date类型对象

Date对象，操作日期的一个工具库

Date对象是JavaScript提供的日期和时间的操作接口。它可以表示的时间范围 是，1970年1月1日00:00:00前后的各1亿天（单位为毫秒）

Date对象可以作为普通函数直接调用，返回一个代表当前时间的字符串。

"Wed May 11 2016 00:31:12 GMT+0800 (中国标准时间)"

注意，即使带有参数，Date作为普通函数使用时，返回的还是当前时间。

Date(2000, 1, 1)// "Tue Dec 01 2015 09:34:43 GMT+0800 (CST)"

上面代码说明，无论有没有参数，直接调用Date总是返回当前时间。

* 了解Date对象的作用
  + 代表日期
* 如何创建对象
  + var d = new Date(); //无参时，括号可以省略
  + var d = new Date("2015/08/22");

//var date = new Date(2015,5,13,12,12,12,998);

//var date = new Date("2015-1-13 12:12:12");

//var date = new Date("2018/1/13");

* 获取设置年月日
  + getFullYear()/setFullYear(2014) 年
  + getMonth()/setMonth(8) 月
    - 注意：获取月份是从0开始的
  + getDate()/setDate() //日
* 获取星期几
  + [getDay()](mk:@MSITStore:D:\\学习资料\\千锋\\尹涛备课资料\\W3School离线手册(2014.3.24).chm::/www.w3school.com.cn/jsref/jsref_getDay.asp) 0-6
* 获取设置时分秒
  + getHours()/ setHours() 时
  + getMinutes() /setMinutes()分
  + getSeconds()/setMinutes() 秒
  + getSeconds()/setSeconds() 秒

getMilliseconds() /setMilliseconds() 毫秒

~~案例：显示当前时间\停留时间~~

* 获取设置毫秒1970年1月1日至今的毫秒

getTime(); / setTime();

* 封装日期处理的函数库
  + ~~判断某年份是否为闰年~~
  + 将日期格式化输出 “2015-08-24”  2016年5月11日 12:12:00
  + ~~获得某个月份的天数~~
  + 将字符串转换为日期
    - join（sep）方法的介绍，将数组按照指定的分隔符转为字符串
    - 例子：

|  |
| --- |
| var arr = ['y','i','n'];  var dstr = arr.join("-");  alert(dstr);//结果 y-i-n |

* + 判断两个日期相差的天数