**JS基础第二周**

**第一天(8月8日)**

**一．数组**

1.引用数据类型和基本数据类型的区别

数据类型：基本数据类型和引用数据类型

基本数据类型：Number，String，Boolean，Null（Object），undefined

引用数据类型：函数，对象，数组

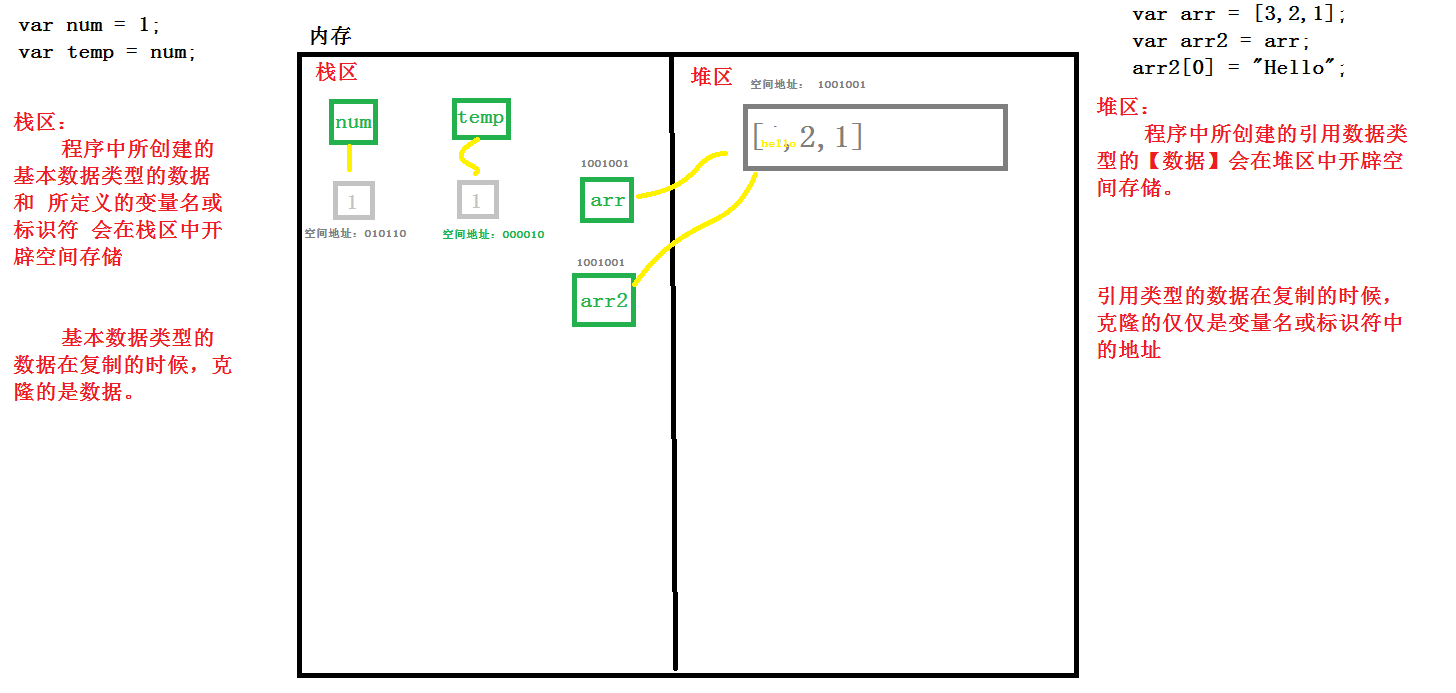
他们的区别在于：

1、对于基本数据类型的数据和变量名以及标识符存储在栈区，

对于引用数据类型的数据存储在堆区；

1. 对于基本类型的数据在复制的时候，会重新对该数据克隆一份，在内存当中。

对于引用数据类型在复制的时候，克隆的是数据和变量名或标识符的地址，而不是数据。



2.数组常用的方法（工具）

Push(数据)：添加一个或多个数据在数组的尾部，并返回当前数组的长度。

Pop（）:删除数组最后一个数据，并吧删除的数据返回。

Unshift（数据）：添加一个或多个数据在数组的首部，并返回当前数组的长度。

Shift（）：删除数组第一个数据，并把删除的数据返回。

Slice（index）：截取，返回一个副本即从索引为index的数据切到最后。不改变原来的数组。

Slice（startindex，endindex）：截取从索引为startindex的数据到索引为endindex的数据。(前包后不包）

Splice（index1，index2）从索引为index1的位子开始删除index2个数据。

Splice（index，1，数据）替换数据

Splice（index，0，数据）添加数据

reverse();反转

sort(function(a,b){})；排序 重点！

一个对象集合如下：

[

{name:"雷哥哥",age:18},

{name:"Bruce.Lee",age:15},

{name:"Andy",age:28},

{name:"Nike",age:19},

{name:"雨哥哥",age:18},

]按年龄排序，若年龄相同则按名字排序，排序完后，把排序的结果打印到页面中

var b = [{name:"雷哥哥",age:18},

{name:"Bruce.Lee",age:15},

{name:"Andy",age:28},

{name:"Nike",age:19},

{name:"雨哥哥",age:18}];

b.sort(function(a,b){

If(a.age==b.age){

Return getValue(a.name)-getValue(b.name);

}

return a.age-b.age

});

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

document.write(b[i].name + b[i].age + "<br/>")

}

Function getValue(str){

Var result = 0;

For(var i = 0;i<str.length;i++){

result+=str.charCodeAt(i);

}

Return result;

}

3. 要求能写：

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)。

var arr = [40,8,15,18,12];

for(var i = 0;i<arr.length-1;i++){

for(var j = 0;j<arr.length-i-1;j++){

if(arr[j]>arr[j+1]){

var temp = arr[j];

arr[j] = arr[j+1];

arr[j+1] = temp;

}

}

}

document.write(arr);

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。

var arr = [3,4,1,5,2];

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

var min = arr[i];

var index = i;

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

if(min>arr[j]){

min = arr[j]

index = j;

}

}

arr[index] = arr[i];

arr[i] = min;

}

document.write(arr);

**JS基础第二周**

**第二天(8月9日)**

**一．字符串**

1. 临时对象

var str = "123456";alert(str.length);//6

首先，将str转换成 对象类型 new String("123456");

然后，对象就可以访问其属性 length;

最后，将结果返回之后，对象释放掉重新转换为 字符串类型

这个过程当中所转换后的对象，被称之为 “临时对象”

str.name = "张三";

首先，将str转换成 对象类型 new String("123456");

然后，给对象添加了一个name属性

最后，这个属性随着临时“对象”的销毁而销毁

这个过程当中所转换后的对象，被称之为 “临时对象”

alert(str.name)//undefined

此时在内存里有会重新创建了一个新的 “临时对象”，这个新的“临时对象” 和 之前被销毁掉的临时对象 是两个不同的独立的 个体.

2. 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

var flag = true; //new Boolan();

alert(flag.valueOf()); //true

alert(typeof flag.valueOf()); //Boolean

alert(flag.toString()); //true

alert(typeof flag.toString()); //String

3. 字符串常用属性（静态属性）和方法（动态属性）

var str = "abcde";

alert(str.length);//5获取字符串的字符数

document.write(str[3]);//d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

alert(str.charAt(3) );//d取一个字符串的指定位置的字符

alert(str.charCodeAt(0) );//97返回一个字符串中指定位置的字符编码

document.write(String.fromCharCode(97) );//编码转换成字符a

document.write(str.indexOf("c") );//2索引查找字符串第一次出现的位置

document.write(str.indexOf("f") );//没有找到为-1

document.write(str.lastIndexOf("c") );//2查找字符串最后一次出现的位置

document.write(str.lastIndexOf("f") );//没有找到为-1

4. search() 正则匹配 (返回出现的位置)

var str="Visit W3School!"

document.write(str.search(/w3school/i))//6

1. match (将匹配的内容存入数组)返回一个数组

var str = "没有情人的情人节";

var reg = /情人/g;

var result = str.match(reg);

document.write(result + "<br/>");//情人,情人

6. 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.(g表示进行全局匹配，i表示匹配的时候忽略大小写)

如：var str = "afbANDNfakfAmf";

var reg1 = /a/g;

var reg = /a/ig;

alert(str.replace(reg1,"恩") );//恩fbANDNf恩kfAmf

alert(str.replace(reg,"恩") );//恩fb恩NDNf恩kf恩mf

7. concat():方法用于连接两个或多个字符串。

8.字符串的截取方法

str.substring(3，5);与slice一样，截取，不改变原数组，前包后不包。

9. 字符串分割

split——根据分隔符、拆分成数组

var c = "你好-他好-大家好-七夕快乐";

alert(c.split("-") );//[你好,他好,大家好,七夕快乐]

10. str.toLowerCase();小写、str.toUpperCase();大写

11. innerHTML改变元素内容。

**三．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); //

**四．Date类型对象**

1. 构造函数，能够创建对象

通过 new 关键字所调用的函数称之为构造函数；

Date对象，操作日期的一个工具库

2. 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。

3. 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");

参数3种形式:2015,5(6月,因为0-11索引),13,12,12,12,998|2015-1-13 12:12:12| 2018/1/13

4. 获取设置年月日 注意：获取月份是从0开始的

getFullYear()/setFullYear(2014) 年

getMonth()/setMonth(8) 月

getDate()/setDate() //日

5. 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 0表示星期天

6. 获取设置时分秒

getHours()/ setHours() 时

getMinutes() /setMinutes()分

getSeconds()/setMinutes() 秒

getSeconds()/setSeconds() 秒

getMilliseconds() /setMilliseconds() 毫秒

7. 获取/设置毫秒1970年1月1日至今的毫秒

getTime(); / setTime();

8. join（sep）方法的介绍，将数组按照指定的分隔符转为字符串

var arr = [2016,8,10];

//var result = arr[0] + "/" + arr[1] + "/" + arr[2];

var result = arr.join("/");

document.write(result);//2016/8/10

9. Date.parse("2015-08-24")//把时间转换成毫秒

转换格式默认支持2015-08-24或2015/08/24

如果需要支持其他格式，只能手动处理

**JS基础第二周**

**第三天(8月10日)**

**一．BOM**

BOM是浏览器对象模型，是Bowser Object Model 的简称，哪浏览器对象模型又是什么概念？

浏览器：我们访问网站，必须通过浏览器这个平台来进行访问，我们之前所写的html、css以及一些js代码也必须通过浏览器来进行解析和运行。所以浏览器就是一个平台，类似一个我们的PC或手机的操作系统一样。

对象： 什么是对象？我们之前所接触的Date就是一个对象，Date对象通过提供给我们一些方法（函数、功能），可以方便的帮助我们操作时间。我们之前讲过，方法（函数、功能）就是一个工具，对象里有很多工具，所以对象我们可以理解为一个小的工具库！

模型：Model，有模式，模样的意思。

所以，通过以上的分析，BOM是什么呢？

就是 浏览器 工具库 说明书！

我们学习BOM就是学习 js 针对操作浏览器所提供的一些 工具

**二．Window对象**

Window 对象

--通过小写的 window 变量引用

--扮演者两个角色

--javascript访问浏览器的窗口的一个接口（入口）

--ECMAScript中的Global对象，全局对象。

什么是全局对象，全局对象就是顶层对象，顶层对象可以理解为CEO。其他的一切全局变量，都是window对象的下属。

注意对于全局变量，我们可以省略window

### Document对象

通过 document 变量引用，我们所有的网页内容都是通过这个Document对象来进行管理的。

比如我们之前学过的document.write();

frames

框架集，类似一个数组，可以通过frames[索引号]（frames[0]）或frames[框架名称]，来获取指定的框架。

### History对象

通过 history 变量可以引用，对于在当前窗口中，我们所访问的所有页面，都会被记录在History对象中。通过这个对象，我们可以很容易的跳转到我们之前所浏览的页面中

history.go(参数);

参数可以是整数数字也可以是字符串；

整数数字，可以是正数和负数，正数向前跳转，负数向后跳转；

~~字符串：可以使所浏览的url中包含的字符串，页面会跳转到距当前页面最近的历史记录页面中；~~

可以用另外两个方法来替 代go方法

history.back();

history.forward();

history.length; 可以获取浏览的历史记录的数量

### Location对象

通过变量 location 引用，location对象可以帮助我们操作url并提供了一下页面导航功能，关于url的操作呢，我们后面再讲。

* + href属性 控制浏览器地址栏的内容
  + reload()方法 刷新页面，有可能会使用缓存（暂存！！！！）
  + reload(true) 刷新页面，不使用缓存（绝对的！）,重新请求服务器。

### Navigator对象

通过navigator对象引用，Navigator对象可以帮助我们检测浏览器的信息以及浏览器所在的操作系统的信息。

* + appName 浏览器名称
  + appVersion 浏览器版本
  + platform操作系统
  + 最新的浏览器已经全面放弃以上这些属性
  + userAgent 用户代理信息，通过该属性可以获取浏览器及操作系统信息

### Screen对象

通过screen对象来引用，可以用它来获取当前屏幕的高度和宽度。

width:获取屏幕的宽度;

height;

h5:视口（view）

### setTimeout(函数，时间（毫秒数）)

超时调用，就是在指定过了多少时间后，执行我们的代码，只执行一次。

清除定时器

clearTimeout(id);

### setInterVal(函数，时间（毫秒数）));

间歇调用，每隔指定的时间，调用一次我们代码，重复调用。

清除定时器

clearInterval();

**JS基础第二周**

**第四天(8月11日)**

**一．DOM**

1、 关系：

ECMAScript是开发语言，BOM是一个大的工具库，可以通过ECMAScript语言调用工具库里的工具，而DOM是BOM大工具库里的一个小的工具库！

2、 什么是DOM？

DOM是文档对象模型（Document Object Model）的简称，它的基本思想是把结构化文档（比如HTML和XML）解析成一系列的节点（对象），再由这些节点组成一个树状结构（DOM Tree）。所有的节点和最终的树状结构，都有规范的对外接口，以达到使用编程语言操作文档的目的（比如增删内容）。所以，DOM可以理解成文档（HTML文档、XML文档和SVG文档）的编程接口。

3、节点的概念

DOM的最小组成单位叫做节点（node），一个文档的树形结构（DOM树），就是由各种不同类型的节点组成。

对于HTML文档，节点主要有以下六种类型：Document节点、DocumentType节点、Element节点、Attribute节点、Text节点和DocumentFragment节点。

浏览器原生提供一个Node对象，上表所有类型的节点都是Node对象派生出来的。也就是说，它们都继承了Node的属性和方法。

DOM节点：

浏览器在解析我们的HTML文档的时候，浏览中的js引擎除了检测我们js程序之外，还会把我们当前的HTML文档中所有标签或文本、属性解析成一个节点，这个节点又叫做对象。每个节点对象都继承了一个原生节点对象Node

4、节点之间的关系

祖先关系、父子关系、兄弟关系

**二、获取节点**

1. getElementById(id) 获取特定 ID 元素的节点

* + window.onload = function () { //预加载 html 后执行

2. getElementsByTagName(tagName) 获取相同元素的节点列表

* + 返回数组，使用[0]来获取

3. getElementsByName() 获取相同名称的节点列表

* + 不是所有标签都有name属性
  + 某些低版本浏览器会有兼容性问题

4. getElementsByClassName() 获取相同class属性的节点列表

* + IE8及以下不支持

// querySelector(“选择器”); //获取单个元素

// querySelectorAll(“选择器”) //获取多个元素，返回数组

**三、获取、设置、删除节点属性**

1、访问元素节点的属性

* + tagName 获取元素节点的标签名
  + innerHTML 获取元素节点里的内容) w3c标准
  + outerHTML/innerText，非 W3C DOM 规范
  + id 元素节点的 id 名称
  + title 元素节点的 title 属性值（鼠标悬停时的提示）
  + className 获取class属性值
  + style CSS内联样式属性值

//var w = window.getComputedStyle(box,null).width; //标准

//var h = box.currentStyle["height"]; //IE8及以下的处理方式

2、getAttribute(属性名) 获取特定元素节点属性的值

* + 获取自定义属性时，**oDiv.xxx** ， IE最新版本不支持，其他浏览器不支持，因此要采用这个方法来访问

常见事件

onload 加载事件网页加载完毕后执行

onscroll 滚动条滚动事件

document.documentElement.scrollTop 代表垂直的滚动条，向下滚动的距离

document.body.scrollTop //chrome 代表垂直的滚动条，向下滚动的距离

document.documentElement.scrollLeft

document.body.scrollLeft

3、setAttribute(属性名，值) 设置特定元素节点属性的值

IE低版本浏览器不支持这个方法

4、removeAttribute(属性) 移除特定元素节点属性

* + 某些低版本浏览器不支持

**四、介绍几个常用的事件**

onload 页面加载完后

onclick 点击

onmousemove 鼠标移动产生事件

onmouseenter 鼠标浮动到某个元素产生的事件

onmouseleave 鼠标离开某个元素产生事件

**JS基础第二周**

**第五天(8月12日)**

1. **案例练习**

实战1： 换肤；

实战2： 随机获取验证码（5个字符）！

实战3： 随机点名！

实战4： 简易年日历！

案例5： 回到顶部！

实战6： 进度条

作业7： 广告弹出框！

1. **Node节点**
2. 常用属性：

**nodeType :表示节点类型**

Document 返回9；Element 返回 1; Text 返回3 ;Attribute 返回 2

常量：值不会发生改变的量叫做常量。

**nodeName：表示节点的标签名**

**nodeValue： 表示文本节点类型的值**

**childNodes** 获取当前元素节点的所有子节点

这里面包含空白节点，在IE9之前，IE浏览器会自动忽略空白节点

children 获取只包含Element类型的节点 和 childNodes类似

**firstChild** 获取当前元素节点的第一个子节点

firstElementChild 返回元素类型为Element的第一个节点

**lastChild** 获取当前元素节点的最后一个子节点

lastElementChild： 获取当前元素节点的最后一个子节点（Element）

**ownerDocument** 获取该节点的文档根节点，相当与 document

**parentNode** 获取当前节点的父节点

**previousSibling** 获取当前节点的前一个同级节点

previousElementSibling 获取当前节点的前一个同级节点(Element)

**nextSibling** 获取当前节点的后一个同级节点

nextElementSibling 获取当前节点的后一个同级节点（Element）

**attributes** 获取当前元素节点的所有属性节点集合

length

childElementCount 子元素的数量 和 length值相等

1. 创建、删除、插入、替换节点

创建：

* + document.createElement(“标签”) //创建一个元素节点(对象)
  + document.createTextNode(“文本内容”) //创建一个文本节点

插入：

appendChild(node)：

box.appendChild(node) //把node节点插入到box的内部最后的位置。

insertBefore(newNode, existNode)

box.insertBefore(newNode, existNode) //把newNode节点插入到exsitNode的前面。

删除：

removeChild(node) 删除某个节点

box.removeChild(node);

替换：

replaceChild(new,old);

box.replaceChild(new,old); 把old替换成new

### 3、offset

* offsetParent（参照物父元素）
  + 当某个元素的父元素或以上元素都未进行CSS定位时，则返回body元素，也就是说元素的偏移量（offsetTop、offsetLeft）等属性是以body为参照物的
  + 当某个元素的父元素进行了CSS定位时（absolute或者relative），则返回父元素，也就是说元素的偏移量是以父元素为参照物的
  + 当某个元素及其父元素都进行CSS定位时，则返回距离最近的使用了CSS定位的元素。