# js周报（第二周）

1. 类是对象的抽象，对象是类的实例。什么是对象，其实就是一种类型，即引用类型。而对象的值就是引用类型的实例。在 ECMAScript 中引用类型是一种数据结构，用于将数据和功能组织在一起。它也常被称做为类，但 ECMAScript6以前却没有这种东西。虽然 ECMAScript 是一门面向对象的语言，却不具备传统面向对象语言所支持的类等基本结构。

对象遍历：for(var key in obj){

    console.log(key);

    console.log(obj[key]);

}

2、定义数组：var arr = [];

//字面量的方式

var arr = new Array();

//构造函数的方式

var arr = new Array(10);//一个参数指数组长度为10

var arr = new Array(10，20，30);//多个参数指定义数组元素

数组遍历：a、for(var i = 0; i < arr.length; i++){

    console.log(arr[i]);

}

b、for(var key in arr){

    console.log(typeof key); //string

    console.log(arr[key]);

}

C、for(var value of arr){

    console.log(value);

}

3、数组常见API：

|  |  |
| --- | --- |
| concat() | 连接两个或更多的数组，并返回结果。 |
| join() | 把数组的所有元素放入一个字符串。元素通过指定的分隔符进行分隔。 |
| pop() | 删除并返回数组的最后一个元素 |
| push() | 向数组的末尾添加一个或更多元素，并返回新的长度。 |
| shift() | 删除并返回数组的第一个元素 |
| unshift() | 向数组的开头添加一个或更多元素，并返回新的长度。 |
| reverse() | 颠倒数组中元素的顺序。 |
| slice() | 从某个已有的数组返回选定的元素 |
| sort() | 对数组的元素进行排序 |
| splice() | 删除元素，并向数组添加新元素。 |
| toString() | 把数组转换为字符串，并返回结果。 |

1. string常见API：

|  |  |
| --- | --- |
| charAt() | 返回在指定位置的字符。 |
| indexOf() | 检索字符串，返回下标 |
| lastIndexOf() | 从后向前搜索字符串。 |
| charCodeAt() | 返回在指定的位置的字符的 Unicode 编码。 |
| fromCharCode() | 从字符编码创建一个字符串。 |
| concat() | 连接字符串。 |
| match() | 找到一个或多个（正则表达式的）匹配。 |
| replace() | 替换与正则表达式匹配的子串。 |
| search() | 检索与正则表达式相匹配的值。 |
| slice() | 提取字符串的片断，并在新的字符串中返回被提取的部分。 |
| split() | 把字符串分割为字符串数组。 |
| substr() | 从起始索引号提取字符串中指定数目的字符。 |
| substring() | 提取字符串中两个指定的索引号之间的字符。 |
| toLowerCase() | 把字符串转换为小写。 |
| toUpperCase() | 把字符串转换为大写。 |
| trim() | 去掉字符串前后空格（ES5） |
| startsWith() | 字符串是否以某个字符开头（ES6） |
| endsWith() | 字符串是否以某个字符结尾（ES6） |
| includes() | 字符串是否包含某个字符（ES6） |
| repeat() | 重复某个字符串几次（ES6） |

5、Math.random()生成一个从0-1（包含0不包含1）的随机小数

生成 min ~ max （min < max）的随机数公式：

    Math.random()\*(max - min) + min

6、javaScript内置对象(十个)

1. Object对象        是所有JavaScript对象的超类(基类)

2. Array对象          数组对象--定义数组属性和方法

3. Boolean对象     布尔对象--布尔值相关

4. Date对象           日期对象--日期时间相关

5. Error对象           错误对象--处理程序错误

6. Function对象     函数对象--定义函数属性和方法

7.Math对象          数学对象--各种数学运算工具(不是构造函数)

8.Number对象       数字对象--定义数字属性和方法

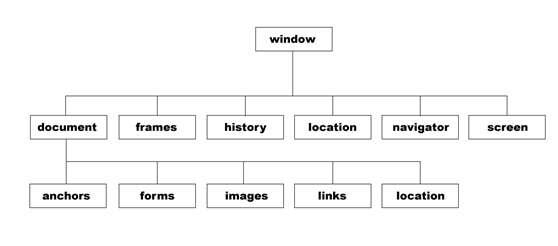
9.RegExp对象        正则表达式对象--定义文本匹配与筛选规则

10.String对象        字符串对象--定义字符串属性和方法

7、Date内置对象

|  |  |  |
| --- | --- | --- |
| getFullYear() | 返回年 | |
| getMonth() | 返回月份0--11 | |
| getDate() | 返回某一天 | |
| getDay() | 返回星期0-6 | |
| getHours() | 返回小时 | |
| getMinutes() | 返回分钟 | |
| getSeconds() | 返回秒 | |
| getTime() | 返回1970年1月1日午夜到指定日期（字符串）的毫秒数 | |
| setFullYear() | 设置年份 |
| setMonth() | 设置月 |
| setDate() | 设置天 |
| setHours() | 设置小时 |
| setMinutes() | 设置分钟 |
| setSeconds() | 设置秒 |
| setTime() | 使用毫秒的形式设置时间对象 |

1. BOM（Browser Object Model  浏览器对象模型）



1. DOM操作

获取DOM节点

   1、 document.getElementById(id名)

 2、getElement**s**ByTagName(标签名) 得到的是一个集合（不止一个，是一堆）

3、getElementsByName( ) 通过Name值获取元素，返回值是集合，通常用来获取有name的input的值； 注：1\*不是所有的标签都有name值；

               2\*低版本的浏览器会有兼容问题；

4、 children属性，获得DOM元素的所有子元素；返回值是集合

 5、parentNode属性，获得DOM元素的父级元素

 6、getElementsByClassName(class名称)

        但是：IE8以下不能用

7、ES5选择器：

 document.querySelector ()  >  一旦匹配成功一个元素，就不往后匹配了

document.querySelectorAll ()  >>>> 强大到超乎想象;匹配到所有满足的元素, 支持IE8+

属性获取和操作

1、getAttribute( )获取元素的属性值，他是节点的方法！所以前缀必须是节点！

    document.getElementById( ID值 ).getAttribute( )

2、setAttribute( )设置元素的属性。同上；

     有些小小的兼容性问题，低版本IE不兼容；

     设置的属性永远都是字符串类型

3、removeAttribute( )删除属性；同上；兼容性问题同上；

操作DOM增、删、克隆节点

创建节点

var oDiv = document.createElement("div");

克隆节点

clonedNode = Node.cloneNode(boolean) // 只有一个参数，传入一个布尔值，true表示深客隆，复制该节点下的所有子节点；false表示浅克隆，只复制该节点

插入节点

parentNode.appendChild(childNode);  // 将新节点追加到子节点列表的末尾

parentNode.insertBefore(newNode, targetNode);  //将newNode插入targetNode之前

替换节点

parentNode.replaceChild(newNode, targetNode);    //使用newNode替换targetNode

移除节点

parentNode.removeChild(childNode);  // 移除目标节点

node.parentNode.removeChild(node);    //在不清楚父节点的情况下使用

childNode.remove()  //IE不支持

10、 DOM尺寸

|  |  |
| --- | --- |
| box.style.width  box.style.height | 只能获取到内联style属性的CSS样式中的宽和高，如果有，获取;如果没有，则返回空 |
| getStyle(box,"width")  getStyle(box,"width")  //如下getStyle方法的封装 | 通过计算获取元素的大小，无关你是否是行内、内联或者链接，它经过计算后得到的结果返回出来。如果本身设置大小，它会返回元素的大小，如果本身没有设置，非IE浏览器会返回默认的大小，IE浏览器返回auto。 |
| box.clientWidth  box.clientHeight | 返回了元素大小，但没有单位，默认单位是px，如果设置了其他的单位，比如100em之类，返回出来的结果还会转换为px像素（不含边框） |
| box.scrollWidth  box.scrollHeight | 获取滚动内容的元素大小（当元素出现滚动条时，此属性指全部滚动内容的宽高）  返回了元素大小，默认单位是px。如果没有设置任何CSS的宽和高度，它会得到计算后的宽度和高度 |
| box.offsetWidth  box.offsetHeight | 返回了元素大小，默认单位是px。如果没有设置任何CSS的宽和高度，他会得到计算后的宽度和高度  包含盒模型中除margin以外的宽高（包含边框）  最稳定，使用最频繁 |

位置坐标

|  |  |
| --- | --- |
| box.clientLeft  box.clientTop | 获取左边框和上边框的宽度 |
| box.offsetLeft  box.offsetTop | 获取元素当前相对于offsetParent父元素的位置 |
| box.scrollTop  box.scrollLeft | 获取滚动内容上方的位置(就是隐藏的内容的高度)  获取滚动内容左方的位置 |