# JavaScript

## 基本知识：

Java的底层语言是二进制，对于人类来说，使用二进制太过于麻烦，人类采用更高级的语言简化编程过程，高级语言由机器翻译成为二进制运行，一般通过浏览器的js引擎直接翻译。这是一门脚本语言，直接运行于客户端而不是服务端（处理器）。它的运行过程主要是逐条解读运行。常见的js输入编写语言有：alert console prompt；

### 1、变量

变量的命名：var 名字 = 值；

变量的使用：alert（名字），不打引号； 变量修改直接改数据或者重新命名覆盖；变量只是我们数据的盒子，数据要自己编写修改。

变量的命名规范要遵守规则：不能以数字和除了￥ -之外的特殊符号开头，一般以完整的英文作为命名的名字。

### 2、数据类型

由于不同的数据所占用的空间是不同的，所以它们被加以区分，分为数字类型和字符串类型。检测数据类型用 console.log(typeof 值) 测试；console.log(名字.length)计算字符串的长度； isNaN（值）可以判断值是不是非数字类型。

console.log(Number.MAX\_VALUE)最大数字类型， console.log(Number.MIN\_VALUE)最小数字类型

在console中可以使用转义符做出一些效果,都是用 \ 开头的，\n 换行，\b空格，\t缩进，不过都要写在单引号内部。

比较常见的几种数据值：null 空值（和字符串相加则为null，和数字相加则为0）； 布尔型有两个值，true false ，代表1 和0； 最后是variable，它的值是 undefined。与字符串相加是本身，与数字相加是NAN（not a number）。

#### 1、数据类型转化

数字转换成为字符型的数据：var 名字 = 被转化值得名字 . tostring(). 或者用console.log(被转化值的名字 + ‘ ’ ) 。

字符型转化成数字型的数据：console.log ( parseInt . （变量） )，得到整数数字；或者得到完整的用 console.log( parseFloat .（变量） )，当然，也可以直接转化，这是测试写法。还可以将字符型的数字用 — + \* / 符号后面加一个 值 把它直接转化成为一个数字型的变量，这种方法是一种隐式转换。

把其它类型转化成为Boolean型，只需要用 Boolean（变量） 就行了，要注意的是，空值，undefined NaN null 0 结果都是false，其它都为true。

## 二、运算符

递增运算符： ++ -- 前置和 后置的区别是，前置先加上 1 赋予原值，后置先赋予原值后加 1 ；

比较运算符：相等 == ； 大于等于 >= ； 全等 === ； 不相等 ！= ；存在隐式转化，会将字符串转化成数值。结果都是false 或者 true。

逻辑运算符：与 && （两个是真才是真）；或 || （一个是真就是真）；非 ！（是真输出就是假反之亦然）。

逻辑中断： 与 &&，当用Boolean值进行运算时，如果第一个表达式是真的，那么输出结果是第二个表达式，如果第一个是假的，那么输出结果是第一个表达式。对于 或 || 来说则是相反的。 当出现逻辑中断的时候，那么中断后面的表达式则不进行计算了。

逻辑简写：num1 -= 1 相当于 num1 = num1 -1 ；num1 += 1 相当于 num1 = num1 + 1

要注意，与 && 的计算优先级要高于 或 || ； 加减乘除的优先级高于比较级

## 三、流程控制代码运算

if语句：三元表达式：var 变量 = 另一个变量 > 值 ？ ‘ 是的 ’ ： ‘ 不是 ’ ；

switch语句：当使用特定值的时候可以使用它，写法：switch （）{ case 值 ：执行代码；break；…… default : 执行的代码 } ；注意 ：break不能省略，值要全等才能被正确执行；

## 四、循环

for 循环：for （变量； 终止条件； 执行代码） { }

对for 循环做断点调试是测试循环是否都正确的方法，具体步骤是在F12中在元素里面断点后按刷新，之后就可以一步步调试；

while循环格式： while （表达式）{ 执行代码 } ； 还有do ... while... 用法需要记忆，先执行do，当while满足，继续执行do。

continue ：跳过特定循环的关键字 ； break：结束循环

## 五、数组

### 1、数组的定义和写法

数组：多个数据的集合体 ；

数组创建：var arr = new Array （）； 或者 var arr = [ ] ;

arr . length 能将数组有几个变量得出来 。搭配循环更好用；

argument [ 0] ；这个是实参数列 ，以后再用多个数列去引用它就能自动生成一个数列，比如function（1，2,3,4,5）最后就能够有一个[1,2,3,4,5]的数列；

判断是否是数组：数组名 instanceof Array；自动判断（instanceof是一个运算符，不是内置对象） ； 或者用 Array.isArray (数组名) 判断是否为数组，返回的值是true 或者false；

### 2、数组追加删除与替换

数组从后面追加元素 ：arr.push(追加值) ； 数组从前面追加值 ： arr.unshift （追加值）；

数组从后面删除一个元素 ：arr.pop ( ) ; 数组从前面删除一个元素：arr.shift ( ) ;

删除指定元素：arr.splice （第几个开始，要删几个数）； 截取数组 ：arr.slice (第几，第几)截取；

要给数组追加元素可以先：arr.length = 值（加长长度） ； 再替换元素：arr [ 第几个] =新的替换值 ； 这样就行了

数组翻转：arr.reverse ( ) ; 数组的排序：arr.sort (function (a , b) { a - b} ) ;

获取数组元素索引号 ：arr.indexOf (元素值，从第几个开始找)，如果等于 -1 那么说明没有该元素

## 六、函数

函数写法：function 名字（ 值，…） { 表达式 } ； 表达式有return，则会把return后面的值返回给整个函数，给人的感觉就像是写了一个var 变量 = 调用值 的样子 ；return 同级的后面代码不会被执行，并且它只能返回一个值 ；

函数内部命名的变量在函数外不能使用；

## 七、作用域

作用域分为全局作用域和局部作用域 还有块级作用域，这里说一下全局变量和局部变量；全局就是函数外命名的变量，局部变量是函数内的变量；全局变量只有在浏览器关闭才会销毁，而局部变量运行完函数后就会销毁，比较省内存

预解析：js在执行代码的时候会先将var 和 function 提到最前面，不过只是提升名字，不提升赋的值； var a=b=c=9 相当于 var a=9 ；b=9 ；c=9；

## 八、**对象**

：对象就是做一个特定事物的所有特征集合，var A = {属性：值 ， 属性，值 …… ; }，还可以在里面加个函数，不过使用的时候的记得 对象.函数（）； 这样用；

调用属性方法 A [ 属性 ] FF1B

创建对象的方法有三种：直接创建对象，用new Object创造对象，构造函数创建对象；

构造函数：它就是对象的集合体：这么写 function Star { this.名字= 值；…… } ，构造函数的名字第一个字母要大写；调用这个函数主要要在函数名前面加上 new 关键字； 而且值得注意的是，构造函数不需要写return这个返回赋值，好像函数也不必须需要喂，艹；

遍历对象要做的是 for （var 变量 in 对象名）{ console.log (对象名[变量]) } 得到对象所有的值； for （var 变量 in 对象名）{ console.log（变量）) } 得到对象所有属性值，这里这个变量我们喜欢写 k 或者key 。

for（var k in 数组 集合对象）{console.log （对象 [第几个] [k]）}可以将特定对象的属性值求出来；

内置对象：也就是那些像vscode里面那些提示语一样的对象

### Math常见对象

Math.abs（值）求绝对值； Math.max（一组值） 求最大值；

Math.min （一组值） 求最小值； Math.PI 这是圆周率

Math.ceil（值） 向上取整 ； Math.floor（向下取整）

Math.round（值）四舍五入取整；

求两个数之间包含两个数的随机数 ：Math.random () \*(max-min +1) +min

求两个数之间包含最小值不包含最大值的随机数：Math. random () \*(max-min ) +min

Date 日期对象

日期对象是一个函数，写法要先引用：var A = new Date（）；

常见的内部各式是字符串型：2023-09-11 09:01:11 格式； 或者数字型 2019,08,13

Date.getYear（）得到年 ； Date.getMonth（）得到月份；

Date.getDay（） 等得星期几； Date.getHours（）得到小时；

Date.getMinutes（）得到分钟； Date.getSeconds（）得到秒；

Date.getDate（） 得到日；

求当前距离1970年1月1日毫秒数： Date.getTime 或者 Date.valueOf（） 或者 console.log（Date.now（）） 或者 var A =+new Date （）；

## 九、特殊知识

字符串：原有的字符串被重新定义和拼接后不会销毁，还是会占内存，所以不能随意拼接字符串和重新赋值一个字符串变量

Warning！！！在给数组[ ]设定js 的function 时，一定要注意循环的长度,不然无法执行！！，而且会浪费大量的时间！！！

<a hre="javascript:;">删除</a> 防止链接跳转！！！

## 十、字符串

str.charAt （值）；查找字符串固定位置的值；

str.charCodeAt（值）返回一个ASCII码；判断用户点击的位置；

str[ 值 ] ； 查找固定位置的值，新版可用；

str.concat（’字符串’） 拼接在字符串后面；

str.substr（开始，结束） 截取字符串，注意位置；

str.slice（开始，结束） 截取字符串，取不到最后一个；

str.substring（开始，结束） 截取字符串，取不到最后一个；

str.replece（被替换的字符，去替换的字符）；

字符串转化为数组：str.split （‘分割符’）；

数组转化成字符型：arr.join (‘分隔符’) ；

## 十一、API

API就是为了使程序员实现某种功能而提供的接口，不必纠结怎么实现的；而web API就是网页交互效果的额及接口，web API 很多都是方法或者是函数；

### 1、DOM（document object model）

文档对象模型，是一种标准程序接口，要执行某个操作，这被看做一个事件，包含三个要素，1、事件源，2、事件类型，3、事件处理程序.

#### 添加事件的方法

1. 用element.onclick = function
2. 用element.addEventListener( click，function(){} )事件监听器来添加；优点是可以反复添加相同事件不同内容，会一个个执行，上一个一次只能添加一个事件；这个是方法，IE9以上支持

#### 解绑事件的方法

1. 用第一种注册事件解绑的方法：在function里面添加点击事件 =null ；
2. 用第二种注册事件解绑，可以考虑将function写在外边然后用调用函数的方法调用该函数，但不需要加（）；解绑就在函数里的执行事件改为null；就行4E86

### （1）、一般的js语句

document.getElementById (‘id名’) ；通过id名字文档获取元素，获取元素所有参数；id 就是给元素取的名字，多数情况下独一无二；

console.dir ( ) ;获取元素的属性或值；

document.getElementsTagName (‘ 元素对象的属性 ’ ) ; 文档通过元素的属性获取元素集合； 返回的是元素对象的集合，是一组伪数组；

当用 属性.getElementTagName （‘属性’）；来获取属性的时候要注意属性是否为伪数组，是的话要指定第几个；

document.getElementsClassName (‘class名字’) ；获取这一类元素的集合；

document.querySelector (‘’) ;也可以根据类名选择相应类名的第一个属性； . 属性 表示根据类，#s属性 表示根据id名，都只能选择第一个哦

document.querySelector All（‘’）, 得到所有；

document.body获取 body 对象; document.documentElement（）；获取整个html 对象；

element.innerText =‘’ ；将元素里面的内容换掉；不过不保留空格和换行；

element.innerHTML = ‘ ‘ ；将元素里面的内容换掉，不过保留空格和换行，并识别html标签；

修改表单的value 值的时候还是得 element.value来进行修改； disabled 禁用

用this.className = ‘’ ; 修改类名的时候，要注意这会覆盖原先的类名；

element.onmouseover = ；当鼠标经过的额时候执行的功能；

element.onmouseout = ；当鼠标离开的时候执行的功能；

#### 自定义属性

自定义属性在不考虑兼容性的问题时，命名规范是data-属性名；自定义属性就是为了存储和使用某些数据，一般这些数据都很简单

element.getAttribute（‘属性名’）；

element.setAttribute（‘属性’） 设置属性；element.setAttribute（‘属性’，‘修改的值’）修改属性值；

element.removeAttribut（‘属性’） 移除属性值

### （2）、节点

节点就是所有html中的各种元素节，他就是用来帮助我们更好的选中元素并操作的；节点主要分为元素节点，文本节点,属性节点

nodeType： 节点类型； 1 为元素节点；2 为属性节点 ；3 为文本节点；实际开发中主要使用的是元素节点

节点.parentNode；找到节点的父级节点；

父级节点.children ; 获得元素所有的元素子节点，不包括文本节点；这还是一个伪数组，是子节点的集合；

父级节点.childNodes ；获得元素所有子节点，包括文本节点

父级节点.firstElementChild ；获得第一个元素节点，IE9以上才支持

父级节点.lastElmentchild ；获得最后一个元素节点，IE9以上才支持

父级节点.firstChild； 获得第一个节点；

父级节点.lastchild；获得最后一个节点；

父级节点.nextSibling ；获得下一个兄弟节点，包括为文本，空格等；

父级节点.nextElementSibling ；获得下一个兄弟元素节点；IE9以上才支持。找不到返回null

#### 创建节点的方法

document.createElement（‘创建的元素’）； //这是第一步

父级节点.appendChild（创建的元素）； //第二步，调用创建的元素

父级节点.insertBefore （创建的元素， 某个节点）； //调用函数插入到某个节点前面

节点.cloneNode（）；复制节点，不包括内容；

节点.cloneNode（true）；复制节点，包括所有内容

复制好的节点要插入才能使用

#### 删除节点的方法

父级节点.removeChild ；删除下一级元素节点；

#### 创建元素的效率

element.innerHTML = ‘’ ；的效率采用拼接字符串的方式效率很低；如果采用数组的形式效率很高

document.createElement（‘’）；的效率挺高，但比采用数组的HTML效率要低一些；

### （3）、DOM技巧

当在 function （）{ }； 在{ }里面写 this时，这个this直接指向函数的调用者；很实用；this 后面跟 .style.属性 = ‘’ ；可以修改元素css 的属性值；更实用；

还有就是事件对象，在function（）里面写上英文打印可以得到该事件相关的信息集合，包括类型或者点击位置等，一般我们喜欢写event或者e或者evt；

### （4）、获取锚点

e.target ；获取被被触发的节点，e指的是事件对象；旧版用e.srcElement ,考虑兼容性

this. ; 获取的是被绑定的对象

### （5）、阻止事件发生

阻止事件发生

e.prevenElement ; 用事件对象阻止事件，e表示事件对象；

e.retrunValue ； 旧版本事件对象阻止事件

return false ； 直接阻止，return后面的代码不执行；

阻止冒泡发生

e.stopPropagation ; 阻止事件向上传播冒泡；

e.canselBubble = true ; 旧版本阻止冒泡；

事件委托： 或者这里提供一个解决办法，我们可以在父级设置一个事件监听器，不再给子节点单独设置监听器，通过父级的程序影响每一个子级，

禁止右键菜单

contextmenu 事件类型 ； e.preventDefaul ; 事件对象属性设置，阻止右键菜单出来；

selectStart 事件类型 ； e.preventDefaul ; 事件对象属性设置，阻止被选中；

获取鼠标点击位置坐标

e.clientX ; e.clientY ; 得到鼠标在浏览器可视区的X ，Y轴的坐标；

e.pageX ；e.pageY ; 得到鼠标在浏览器页面的X，Y轴的坐标

e.screenX ；e.screenY ；得到鼠标在电脑屏幕的X，Y轴的坐标；

mouseover 事件类型，鼠标经过；

mousemove 事件类型 鼠标移动

### （6）、按键发生事件类型

onkeyup 键盘弹起发生 ；

onkeydown 键盘按下发生 ；

onkeypress 键盘按下发生 ，不能是被ctrl shift 键；

执行顺序： onkeydown---onkeypress--- onkeyup ；

在事件监听器中不要加 on ；

每一个按键都有自己的ASCLL值，我们可以通过ASCLL值来判断用户具体按下了那个键；e.keyCode 得到ASCLL值；keyup 和 keydown是不区分大小写的，但能够识别所有的功能键，keypress区分大小写，大小写返回不同的ASCLL码值；

函数里面获得焦点这样写：Element.focus（）；

### 2、BOM（browser object model）

浏览器对象模型，包含整个浏览器；它的最高级对象是window，主要作用是和浏览器窗口做一些交互效果，比如后退页面，拉伸，前进页面等；

#### （1）、一般BOM语句

页面加载完之后出现的事件；

window.addElementListener (‘load’ , function(){} )，在整个页面加载完之后会执行的事件，

或者 ，元素. addElementListener (‘DOMcontentLoaded’ , function(){} ) 在DOM 加载完毕后就会执行的事件，不包含图片，flash，css等东西；

注意，使用的时候无限套娃就对了！

页面尺寸变化会执行的事件

onresize 事件类型，尺寸变化；

window.innerWidth 获得窗口屏幕长度值；

#### （2）、定时器

window.setTimeOut (function(){}，毫秒数) ；定时器可以做一些页面动态效果；因为是window开头，window可以省略；一般我们在设置定时器的时候要给他设置一个标志符，var 命名 = setTimeOut这样的； 这个也是一个回调函数；

window.clearTimeOut( timeID) ；停止计时器，跟alert的使用差不过，这个timeID就是我们给计时器起的名字，清除的是settimeout的定时器

window.setInterval. (function(){}，毫秒数);每隔一段时间调用一次函数；

window.clearInterval（timeID）清除上面这个重复执行的定时器；

回调函数，在某件事干完之后执行的函数；

在js中，所有的回调函数都属于异步任务，其它的为同步任务，异步任务执行会放到后面执行；所以会出现一些延迟的效果；而js 中有一个重要的机制事件循环，就是会不断的去异步任务中找是否存在任务需要执行，这个过程叫做事件循环；

#### （3）、URL（uniform resource locator）

标准资源地址，

location.href 获取得整个URL；

location.search 获取得参数；

location.assign (‘ 跳转的地址 ’); 加载到某个页面；记忆了上一个页面，可退回；

location.replace （‘ 跳转的地址 ’） 加载到某个页面；不记忆上一个页面，不可退回；

location·reload （）； 刷新，当（true）时为强制刷新，相当于ctrl+F5；

navigator.userAgent ；返回客户机user-agent头部值；运用于识别用户是否使用的是移动端还是客户端；

history.back（）； 后退一个页面；

history.forward （）； 前进一个页面；

history.go( 1 )；数值表示前进的页面数；当为-1的时候，为后退一个页面；

#### （4）、偏移

element.offsetParent（）；返回带有定位的父级元素；

element.offsetLeft（）；测量距离浏览器左侧的距离；内部盒子测量时是距离带有定位的父级盒子距离，否则就是距离body的距离；

element.offsetTop（）；测量距离浏览器顶部的距离；内部盒子测量时是距离带有定位的父级盒子距离，否则就是距离body的距离；

element.offsetWidth；返回元素的宽度；包含margin，padding，和border；

element.offsetHeight；返回元素的高度；包含margin，padding，和border；

这两个属性和style.width与style.height 的区别在于1：上面两个是只读属性，不可更改，只能得出一个值，而style里的长宽是可更改的； 2：上面两个包含margin的值，而style不包含； 3：上面的可以得到所有样式里面的长宽，而style只能得到行内样式表中的长宽值； 4：得到长宽值最好用offset，修改长宽则用style ，这样好用一些；

element.clientTop ; 得到元素上边框的长度；

element.clientLeft ；得到元素左边框的长度；

element.clientWidth ; 得到元素的宽度，不包含边框；padding；

element.clientHeight ；得到元素的高度,不包含边框，包含padding ；

这两个和off系列的区别在于它们不包含边框的长度，而off系列包含；

#### （5）、立即执行函数

顾名思义，当有js，立即执行，并且里面的变量都属于局部变量，不会影响外边的变量命名；

写法1：（function（） {执行函数}（调用变量））

写法2、（function（）{执行函数}）（调用变量）；

#### （6）、获取函数物理像素比

window.devicePixelRatio；获取物理像素比；这是在屏幕显示端呈现的像素比大小，像素为1像素比是2则看到的像素就是2像素，懂了吧；

DomContenLoaded ，事件类型，页面全部加载完毕；

事件类型：pageShow就是页面重新加载后执行的事件，pageShow和Load的区别在于，在老版本火狐浏览器中，返回一个页面时因为火狐具有某些保存功能，导致返回后刷新的页面不能执行load的功能，所以开发了一个pageShow来补全这个bug，现在的火狐已经把这个问题解决了，

element.scrollTop ；返回被滚动条卷曲的元素上部距离，不包含单位；

element.scrollLeft；返回被滚动条卷曲的元素左侧距离，不包含单位；

element.scrollWidth；返回滚动条实际宽度，不含边框；

element.scrollHeight；返回滚动条实际高度，不含边框；

window.pageYOffset；返回整个页面被滚动条卷曲的上部长度；

window.pageXOffset；返回整个页面被滚动条卷曲的左部长度；

小总结：element.offsetWidth；element.clientWidth；element.scrollWidth 的区别在于，第一个包含值边框，限于可视区；第二个不包含边框，限于可视区；第三个不包含边框，但不限可视区，比内容较大超出盒子时可以测量盒子内容的实际大小；

事件类型mouseover内的子盒子被鼠标经过时也会触发事件，因为有冒泡情况的发生，解决这个问题可以用mouseenter，这样经过子盒子就不会发生冒泡触发事件多次的情况发生了；同样，mouseleave也能防止冒泡；