**前端之JAVAScript**

**一：JAVASCript概念：**

**简称JS，专门用来编写交互程序的语言。**

**DOM标准：专门操作网页内容的标准，所有浏览器都兼容的标准。**

**BOM ：专门操作浏览器窗口的编程接口。**

**JS = ECMAScript + DOM标准 + BOM**

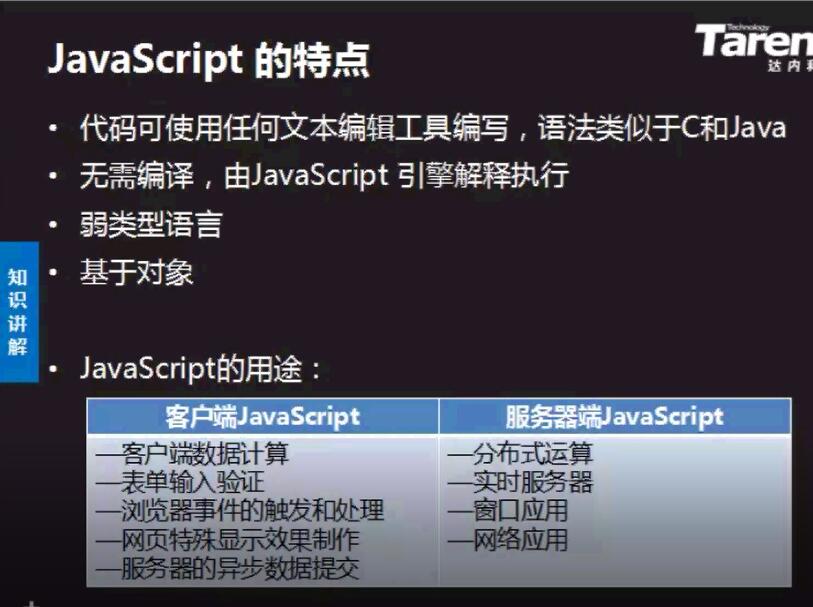
**注：ECMAscript是核心，也是重点。**

**前端三大语言：HTML CSS JS**

**HTML:专门编写网页内容的语言。**

**CSS ：专门美化网页内容的语言。**

**JS :专门用来编写交互程序的语言。**

****

**二：如何使用JS：**

**如何运行：2种**

**1.浏览器内置JS引擎**

**2.独立安装JS引擎：nodeJS**

**三：如何编写：4种方式**

**1. 直接在浏览器控制台（F12）中编写JS脚本。**

**about:blank 空白网页。**

**Console：是控制台，控制台是专门编写和调试JS程序的**

**注意：程序中的关键字区分大小写！！！**

**但是字符串中的字符不要求大小写！**

**2个按键**

**多行程序：shift+Enter（在控制台写JS程序时）**

**上下方向键：切换已经执行过得代码，避免重复编写。**

**2. 绝大数的在网页中编写JS脚本程序：有3处**

1. **<script>元素中**

**直接编写在<script>中的js程序，随网页加载过程执行。**

**Document:当前网页。**

**Write:写入功能，直接写入body元素中。**

**不带换行**

**Write中的字符串必须符合HTML语法。**

**强调：write中换行：<br>**

1. **单独js文件保存js脚本程序：2步**

**<1>.创作。JS文件，保存JS脚本程序**

**<2>.<script src=”url”></script>引入外部监事文件。**

**注意：脚本程序：不需要预编译，边解释边执行的程序。**

**3. 元素的时间处理程序中编写------（此部分了解）**

**不随网页加载过程执行，只有出发时间时才会执行。**

**Window : 指代当前窗口**

**alert : 弹出警告框**

**注意 ：document.write不常用，以后调试或者测试程序时，建议大家用window.alert或console.log**

**五：如何调试**

**Bug:程序中出现的一切错误。**

**Debug:发现错误，解决错误。**

**JS程序出错：**

**希望的交互效果没有出来->F12->console**

**Console:错误信息：有3部分内容：**

**错误类型，错误的原因，出错位置的链接**

**出错现象：出错位置同，<script>下的程序：**

**出错钱的程序正常执行，出错位置以及之后的程序不会执行，其他<script>中的程序不受影响。**

**六：变 量**

**概念：内存中存储1个数据的存储空间，再起个名字。**

**程序中的数据都要先用变量保存，再处理。**

**引申：JS中的变量 和数据库的区别？**

**变量是内存中的，是临时的，是客户端的；数据库是硬盘上的文件，是持久的，是服务器端只存一份。**

**如何使用变量：先声明，再赋值，最后取值。**

**如何声明：Var 变量名;**

**注意：强烈建议声明的同时初始化。**

**强调: 仅声明未初始化变量，初始值为undefined**

**命名规则：**

1. **不能数字开头**
2. **不能使用保留字**
3. **建议命名见名知意，最好用英文**
4. **驼峰命名：首字母小写，之后每个单词首字母大写 eg:listStyleColor**

**取值：从变量中取出数据，进行运算。**

**如何取值：在程序中任何位置，只要使用变量名等效于直接使用变量中存储的数据。**

**特殊情况：2种**

**（1） 为一个从未声明过的变量赋值，不会出错，会自动创建同名变量再保存。**

**（2） 尝试从一个未声明过的变量中取值，会出错**

**注：ReferenceError：未找到或未声明**

**七：常量**

**一旦声明并初始化后，值不可更改。**

**When:只要保存一旦创建，不可擅自改变的量时。**

**比如：g**

**How：const 常量名=值；**

**常量名全大写**

**八：数据类型**

**What：内存中存储不同数据格式**

**JS中数据类型：2大类**

1. **原始类型：值保存再变量本地的数据类型。**

**5种：（1）Number 专门保存数字的类型**

**所有数字都占有8字节**

**1GB=1024MB**

**1KB=1024bytes**

**1byte=8Bit**

**（2）String 专门保存字符串的类型**

**内存： js的程序内存中的字符都是用unicode标示的。**

**Unicode:对全球主要语言中的每个字符都编一个号。**

**所占空间：每个字母，标点占1个字节**

**每个汉字占2个字节**

**字符串一旦创建不可更改！**

**要想改变字符串只能创建新的字符串，替代旧字符串。**

**(3）Boolean 专门保存真假二选一的类型**

**（4）Undefined 只有一个值undefined**

**（5）Null 表示不指向任何地址**

1. **引用类型：值不保存在变量本地的数据类型。**

**九：数据类型的转换**

**JS是弱类型编程语言：3方面**

1. **声明变量时，不用规定变量的存储的数据类型。**
2. **赋值时，动态决定变量的数据类型，同一个变量可先后保存不同的类型的数据。**
3. **运算时，JS会根据需要，动态转换数据的类型**

**2大类：**

**隐式转换：**

**无需程序员干预，JS自动完成的类型转换**

**仅讨论+运算中的隐式转换：**

**算数计算中，一切类型都隐式转换味numberl类型，再计算**

**比如：”2”->2 true->1 false->0**

**特例：+运算中只要有一方是字符串，两类型都转化为字符串类型，且+运算变为字符串拼接！**

**表达式：有数据，变量和运算符组成的一个公式，每个表达式默认都从左向右，两两计算，每个表达式有且仅有一个运算结果。**

**隐式转换，仅仅影响表达式的运算结果，不影响变量中存储的实际类型。**

**强制转换：程序员通过主动调用专门函数，执行转换。**

**1.任意类型转string:2种**

**var str=x.toString();**

**JS种，一切数据都有toString方法。**

**Var str=String(x); ->隐式转换的原型**

**String->Number :2种；**

**任意类型 ->Number: var n=number(x);**

**String->Number :2种；**

**Var n=parseInt(str);**

**原理：从str来时位置逐个读取每个字符，直到碰到第一个不是数字的字符时，停止读取，自动忽略开头碰到的空格。**

**Number VS parseInt(str)**

**比如：Number（“12px”)->NaN**

**parseInt(“12px”) ->12**

**var n=parseFloat(str);**

**原理和parseInt完全一致，但是parseInt 认第一个小数点。**

**十：运算符和表达式**

**1.程序 ：让计算机按照人的想法去执行任务。**

**运算符 ：程序中模拟人的思维运算或判断的符号。**

**两字符串做比较：一次PK每一位字符的unicode编号，只要有一位字符，分出大小，就不再比较。**

**比如:“3”>“10”->true**

1. **NaN : NAN和任何数据座大小或等于比较永远返回false**

**NaN和任何数据作大小或不等于比较，永远返回true**

**isNaN(number(“”));->false 因为number(“”)可以将空字符串转化为数字0；**

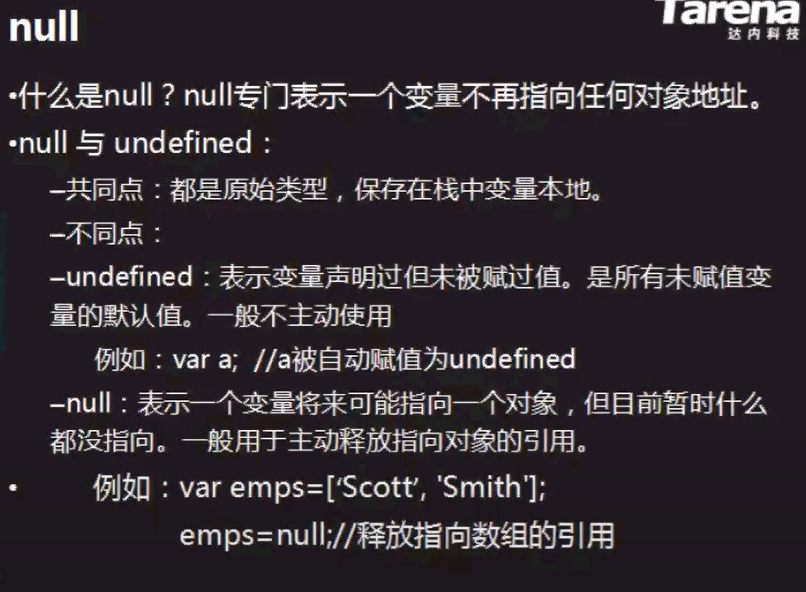
1. **undefiend VS null**

**undefined:是所有没有值得变量的默认值**

**Null ：null专门表示一个变量不在指向任何对象地址。**

**主动释放一个变量引用的对象**

**何时使用null? 当使用完一个呼叫大对象是，主动释放对象总是一个好的对象。**

****

1. **如何判断一个字符是汉字？**

**第一个汉字是“一”->”\u4e00”**

**最后一个汉字“\u9fa5”**

1. **位移运算：2个**

**左移：n<<m 将n左移m位。**

**右移：n<<m 将n的二进制，右移n位。**

**0 0 0 0 0 0 0 0**

**128 64 32 16 8 4 2 1**

1. **三目运算：目：指的是参与运算的个数。**

**五：数组：**

**5.1.1概念：**

**一组持续的变量组成的集合--统一起一个名字批量管理多个数据。**

**5.1.2 如何使用？ 创建 赋值 取值**

1. **创建：4种（注意：后两种不常见，不常用,了解一下即可）**

**1. var 变量名=[]; -->创建一个空数组**

**2. var 变量名=[值1，值2，值3，...] -->创建一个数组的同时给数组指定初始元素（数组中每个值）。**

**数组是引用数据的对象。**

**变量中保存了数组对象的地址，也称引用了数组对象。**

**3. var arr= new Array(); -->创建一个空数组对象。**

**new : 创建一个新对象，并返回新对象地址。**

**Array : JS中的数组类型**

**New Array（） -->创建一个数组类型的新对象，并返回一个新对象的地址。**

**4. var arr=new Array(n); -->创建n个元素的数组**

**另一种写法：new Array(值1，值2，...)**

**5.1.3 如何找到数组：**

**找到数组：使用变量等效于直接使用变量引用的数据对象。**

**数组对象中，每一个元素都有一个下标：每一个元素的位置号下标从0开始，到元素个数-1结束**

**5.1.4 ！！！ JS数组 VS 其他数组**

**<1>JS中数组越界不会报错！！！**

**<2>为不存在的位置赋值，不会出错，会自动创建指定下标的新元素。**

**<3>从不存在的位置赋值：不会出错！也不会增加新元素！而是返回undefined**

**5.1.5关联数组：**

**可以自己定义下标名称的数组**

**1. 为什么? [“范冰冰”,91,67,95]**

**Sname mach yuwen eng**

**2. 如何创建关联数组：var fbb=[];**

**Fbb[“sname”]=”范冰冰”;**

**Fbb[“mach”]=91;**

**3. 如何访问关联数组中的元素？**

**Fbb[“sname”];**

**关联数组中的.length属性失效！**

**关联（hash）数组：下标是不能重复的**

**优势：利用Hash算法，精确定位某个下标的位置**

**不用遍历。**

**索引数组：缺：下标自动分配，无意义**

**要想按内容查找元素，只能从头开始遍历**

**遍历关联数组：**

**For(var key in arr)**

**5.1.6原始类型 引用类型**

**原始类型赋值修改其中一个对象，另一个变量不受影响**

**eg : var m=100;**

**n=m;**

**n++;**

**Console.log(m); //输出结果为100**

**而引用类型对象变量间赋值，只要任意一方修改对象，另一方变量受影响。**

**eg . var my=["包子","包子","包子","包子","包子"];**

**var lp=my;**

**Lp[0]=0;  
console.log(my); //剩下4  
my[1]=0;  
console.log(lp); //剩下3**

**5.1.7 undefiend VS null**

**undefined:是所有没有值得变量的默认值,一般自动赋值**

**Null ：null专门表示一个变量不再指向任何对象地址。**

**主动释放一个变量引用的对象**

**何时使用null?**

**当使用完一个呼叫大对象是，主动释放对象总是一个好的习惯。**

**垃圾回收器:专门释放对象内存的一个程序**

**在底层，在后台，伴随当前程序同时运行**

**引擎会定时自动调用垃圾回收器。**

**总有一个对象不在被任何变量引用时，才会释放。**

**5.1.8**

**数组是对象：封装了一组数据，并提供了对数据的操作方法。**

**.length属性：获得数组中的元素个数！=实际的个数**

**.length属性：只是象征意义。**

**何时使用：3种**

**（1）arr[arr.length-1]:获得任意长度数组中的最后一个元素。**

**（2）arr[arr.length]=新值：向数组末尾追加一个新元素。**

**改小length的值，可删除末尾元素。**

**数组遍历：从下标0位置开始，依次取出每个元素，反复执行相同的操作。**

**三要素：**

**1.循环条件：下标<arr.length**

**2.循环变量：下标i,从0开始，每次增加1，到arr.length-1结束。**

**3 循环体 ：xxx**

**十一：**

**API:应用程序编程的接口**

**已经实现的，现在的对象和方法。提高开放效率。**

**引申：判断输入的一个字符是否是汉字：**

**如果“\u4e00”<=字符<=“\u9fa5”,输出是汉字**

**三：函数：**

**3.1概念：封装一个专门任务的步骤清单的代码段起一个任务名。**

**3.2何时使用：只要定义一个专门的任务，都要封装一个函数。**

**使用函数：声明 定义 和调用**

**如何声明并定义一个函数：**

**Function 函数名([参数变量列表]){**

**函数体；**

**}**

**函数只用调用时才会执行，反复调用，就会反复执行相同的操作！**

1. **Tofixed(数值) 保存到xxx位小数**
2. **CharCodeAt()**
3. **Prompt()**
4. **isNaN(n) 专门判断n 是否是数字。**

**！！！在任意min-max之间取随机整数的公式：**

**parseInt(Math.random()\*（max-min+1）+min)**

**0<=xxx <1**

**1.\*\*\*函数：**

**注意：函数声明提前不是很懂！！！**

**\*声明提前： 在程序执行前或函数被调用前，将var生命的变量和function声明的函数提前到当前作用于的顶部集中闯将**

**强调：仅声明提前，赋值留在原地。**

**\*\*\*按值传递：JS 中无论变量赋值或使用变量传递参数时，都是将变量中的值，复制一个副本给对方!**

1. **全局函数（作为了解）**

**就是ES标准中已经定义好的，开发中可以直接调用的函数。**

**parseInt/Float(str)**

**isNaN(num)**

**编码:将url中的非法字符，改为合法字符表示**

**http://.../s?word=%E5%BC%A0**

**张**

**Utf-8格式编码**

**汉字占3字节**

**url中不允许出现多字节字符！**

**解决：将url中的多字节，变为单字节。**

**URL中不允许参数值中出现保留字符：比如：/**

**http://tmooc.cn-->http%3A%2f%2ftmooc**

**encodeURI()**

**解码：将url中的非法字符编码后的内容，恢复成原文decodeURI()**

**十二：Web 浏览器中的JAVAScript**

**概念：通常称为客户端的JavaScript**

**12.1客户端的javascript**

**window对象是所有客户端的javascript特性和API的主要接入点。它表示浏览器的一个窗口或窗体。并且可以用标识符window来引用它。**