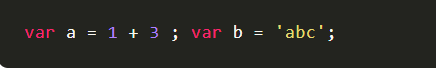
导语：js是一行一行的执行命令，语句以分号结尾，区分大小写，动态语言类型，变量可以随时改变类型（整型或字符串等）

#### 1.语句

Var 声明一个变量 var 变量名 = 值（可放各种表达式 如 a+b，’abc’）



分号前没内容视为空语句



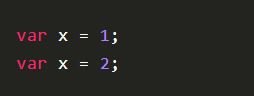
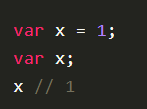
#### 2.变量

变量的声明和赋值可分开写

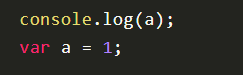


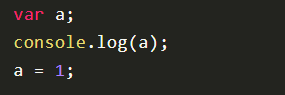
如果只是声明变量而没有赋值，则该变量的值是undefined。undefined是一个特殊的值，表示“无定义”。

如果变量赋值的时候，忘了写var命令，这条语句也是有效的。同一个变量多次声明是无效的，第二次及以后的无效，但若声明后赋值则会覆盖前面的值



用console.log的办法在控制台显示a的值 console（控制台），上述代码不会报错，但是错误的，因为变量在打印时还未声明赋值，所以不会显示（语句是一行一行的运行）



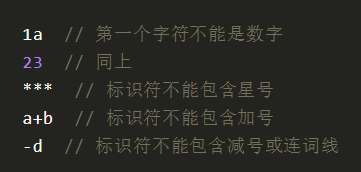
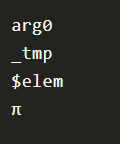


只会显示未定义（undefined）

#### 变量名命名规则

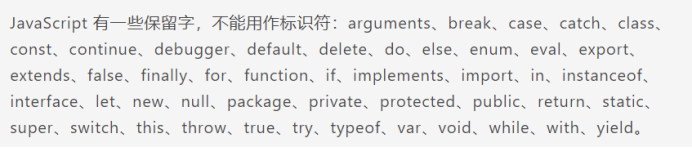
1）.第一个字符，可以是任意 Unicode 字母（包括英文字母和其他语言的字母），以及美元符号（$）和下划线（\_）。

2）.第二个字符及后面的字符，除了 Unicode 字母、美元符号和下划线，还可以用数字0-9。



中文也是合法的变量名

IMG_265



这些具有特殊意义，不可做变量名

注释：//单行注释 /\* \*/多行注释 html语句中<！--和-->在js中也合法其中-->放在首行才表示注释，否则会被当作运算值

IMG_267

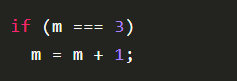
只有x =1 可以被执行

{}表示一个作用域（scope），但var命令一般不单独构成作用域，如

{var a = 1；var b ；b = 3 ；}

作用域一般用于复杂语句如函数function，if，while条件语句，for循环

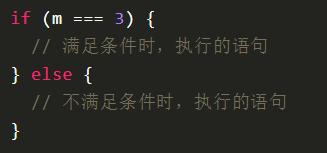
If（条件）{执行什么语句}

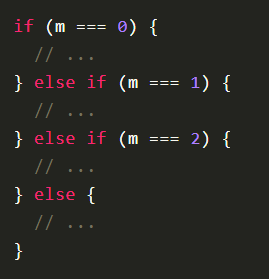


赋值表达式（=）、严格相等运算符（===）和相等运算符（==）

我们一般都习惯在后面加上{}，这样代码逻辑更清晰，也方便插入其他语句，条件成立执行{}内语句，不成立执行{}外语句

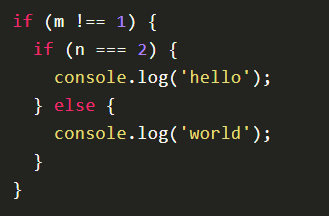
If....else结构

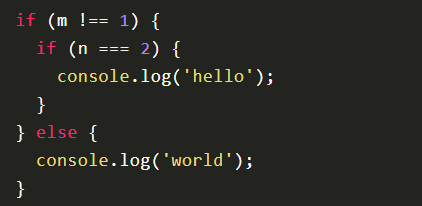




多个if... else连写

Else会与自己左边最近的}所指向的if配对





两者else匹配的不一样

## Switch

多个if else连用时，可以用switch替换

switch（某个变量名或变量值）{

Case 变量名或值的条件：

Console.log()； //输出

Break;//break必写，不然无法跳出循环

Case 变量名或值：

Console.log();

Break；

Default：//条件都不满足时，执行语句

Console.log();

}

#### 三元运算符

三元运算符（即该运算符需要三个运算子）?:，也可以用于逻辑判断。

(条件) ? 表达式1 : 表达式2

若条件成立返回表达式1，不成立返回表达式2

可理解为if... else的缩写

var even;if (n % 2 === 0) {

even = true;

} else {

even = false;

}

#### 循环语句

##### While循环

while (条件) {

语句;

}

条件成立语句执行，不成立停止执行

For循环

for (初始化表达式; 条件; 递增表达式) {

语句

}

初始化表达式（initialize）：确定循环变量的初始值，只在循环开始时执行一次。

条件表达式（test）：每轮循环开始时，都要执行这个条件表达式，只有值为真，才继续进行循环。

递增表达式（increment）：每轮循环的最后一个操作，通常用来递增循环变量。

例：var x = 3;for (var i = 0; i < x; i++) {

console.log(i);

}

##### Do....while循环

do {

语句

} while (条件);

不管条件是否成立都会循环一次，循环结束为条件不成立时

##### Break和continue

**Break是终止循环，continue是跳过本轮循环，如**

for (var i = 0; i < 10; i++) {

if (i === 3){

continue;

}

if (i === 8){

break;

}

console.log(i);

}

输出为

0 1 2 4 5 6 7跳过第三次循环，执行第四次循环，终止在第八次循环

#### ****标签****

语句的前面有标签（label），相当于定位符，用于跳转到程序的任意位置

标签结构

label:

语句

Label 变量名由自己定义

用法 例

top:

for (var i = 0; i < 3; i++){

for (var j = 0; j < 3; j++){

if (i === 1 && j === 1) break top;

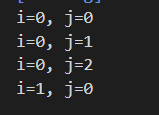
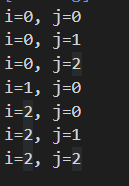
console.log('i=' + i + ', j=' + j);

}

}

当if条件达到时，结束整个top标签语句，当去掉top时，只结束当前所在循环，即第二个for循环，第一个for循环任然执行

去掉top输出： 未去掉top：



去掉top，break停止的是第一个for循环i= 1；的条件下的循环，故i= 2 的循环仍会进行

标签与continue语句使用时，标签相当于并未起作用，只有continue起作用了

top:

for (var i = 0; i < 3; i++){

for (var j = 0; j < 3; j++){

if (i === 1 && j === 1) continue top;

console.log('i=' + i + ', j=' + j);

}

}// i=0, j=0// i=0, j=1// i=0, j=2// i=1, j=0// i=2, j=0// i=2, j=1// i=2, j=2