**二、javascript流程语句**

**一、程序的三大流程控制**

**1.流程分类**

(1)顺序语句 自上而下顺序执行

【例】输入两个数，然后交换这两个数，再输出它们交换后的结果。

var num1 = 10;

var num2 = 20;

var tmp = num1;

num1 = num2;

num2 = tmp;

alert("num1:" + num1 + ", num2:" + num2);

(2)条件/分支(选择)语句

(3)循环语句

**二、条件语句**

**1.单分支语句**

【格式】

if(判断的条件)

只能决定紧跟着的这一条语句;(如果判断条件为true，执行这条语句，否则，不执行这条语句)

【注】JS，如果多条语句，由大括号括起来，大括号所括起来的所有语句，是当做一条语句去对待的。

【标准格式】

if(判断的条件){

当判断条件为true，执行的代码;

}

【例】闰年：1、能被400整除，2、或者<1>能被4整除且<2>不能被100整除的，这个是闰年。

var year = 2000;

if(year % 400 == 0 || year % 4 == 0 && year % 100 != 0){

alert("这是一个闰年");

}

**2.双分支语句**

if(判断条件){

条件为true，执行的代码;

}else{

条件为false，执行的代码;

}

【例】判断一个数的奇偶性。

var num = 7;

if(num % 2 == 0){

alert("这是一个偶数");

}else{

alert("这是一个奇数");

}

**3.多分支语句**

【格式】

if(判断条件1){

判断条件1为true，执行代码

}else if(判断条件2){

判断条件2为true，执行代码

}

...

else{

上述所有条件都不成立的时候，执行的代码;

}

【例】分段函数

var x = 5;

var y = 0;

if(x < 1){

y = x;

}else if(x >= 1 && x < 10){

y = 2 \* x + 1;

}else{

y = 5 \* x - 17;

}

alert(y);

**4.switch多分支语句**

【格式】

switch(表达式){

case 常量1:

break;

case 常量2:

break;

...

default:

//在上述所有条件不成立的情况下，执行

break;

}

【注】执行效率上讲，switch语句要比if语句的执行效率高很多。

【注】与if语句的区别：

1、if(判断条件)，case常量

2、if逐级进行判断，case直接可以去执行对应的选项

【注】使用switch的时候需要注意

1、break一定不能省略，如果不写，整个代码会被穿透。

2、default必须写，防止错误操作。可以在default中找出异常。

【注】

1、如果分支很多，并且分支都是确定的值，选switch

2、否则选if语句。

【例】

var grade = "A";

switch(grade){

case "A":

alert("80-100");

break;

case "B":

alert("70-79");

break;

case "C":

alert("60-69");

break;

case "D":

alert("<60");

break;

default:

alert("输入错误");

break;

}

**5.条件运算符/三目运算符**

表达式1 ? 表达式2 : 表达式3;

【执行顺序】

<1>先去判断表达式1

<2>如果表达式1为真，执行表达式2

<2>如果表达式1为假，执行表达式3

【例】判断两个数中的最大数

var num1 = 10;

var num2 = 20;

var max = null;

if(num1 > num2){

max = num1;

}else{

max = num2;

}

// var max = num1 > num2 ? num1 : num2; //改写成三目运算符

alert("最大值是:" + max);