一.运算符

4.位运算符(了解)

模拟计算机底层的运算，先把值转为2进制，然后进行运算，运算完再把结果转为10进制

1 2 3 4 5 6 7

1 10 11 100 101 110 111

& 按位与，上下两位比较，都是1结果是1，否则是0

| 按位或，上下两位比较，含有1结果是1，否则是0

^ 按位异或，上下两位比较，不同是1，否则是0

>> 按位右移，删除最后的位数，每移动1位是原来的一半，如果是小数进行取整

<< 按位左移，在末尾补0，每移动1位是原来的2倍

7>>2 7<<2

111 111

11 1110

1 11100

3&5 5&7 6|12 7^15

011 101 0110 0111

101 111 1100 1111

——— ——— ——— ———

001 101 1110 1000

5 14 8

5.赋值运算符

= += -= \*= /= %=...

赋值 运算赋值：先执行运算再执行赋值

练习：声明变量保存商品的价格，对价格执行打八折，最后打印结价格

6.三目运算符

一目：由一个运算符连接了一个操作数据或者表达式 ++ -- !

二目：由一个运算符连接了两个操作数据或者表达式

三目(条件运算符)：由两个运算符连接了三个操作数据或者表达式

条件表达式 ? 表达式1 : 表达式2

如果条件表达式是true，执行表达式1

如果条件表达式是false，执行表达式2

二.浏览器端函数

alert() 弹出警示框

prompt() 弹出提示(输入)框，需要使用变量保存用户输入的值，类型是字符串型，点击取消结果是null

练习：弹出两次提示框，分别输入数字；计算两个数字相加的和，并将结果以警示框形式弹出。

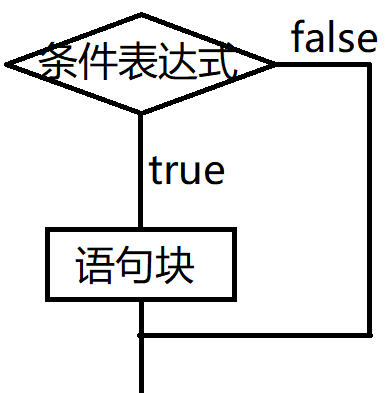
程序 = 数据 + 算法

程序的执行方式：顺序执行、选择执行、循环执行

三.流程控制

1.if语句

满30减20

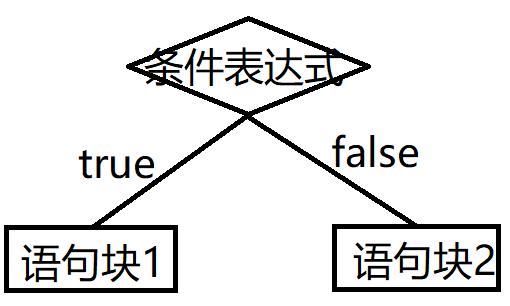


|  |
| --- |
| if(条件表达式){  语句块  } |

如果if后的语句块中只有一行代码，则大括号可以省略的

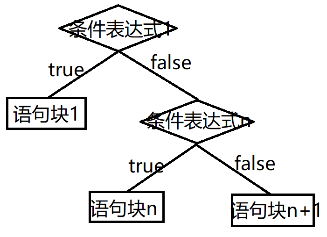
以下5种情况隐式转换为布尔型是false：0 NaN '' undefined null

2.if-else语句



|  |
| --- |
| if(条件表达式){  语句块1  }else{  语句块2  } |

3.if-else嵌套



|  |
| --- |
| if(条件表达式1){  语句块1  }else if(条件表达式n){  语句块n  }else{  语句块n+1 //以上所有的条件都是false  } |

4.switch-case语句

是一种特殊的多项判断语句，只能进行全等于(===)一种比较

|  |
| --- |
| switch(表达式){  case 值1: //如果表达式和值1相等  语句块1  break; //结束整个switch语句  case 值n:  语句块n  break;  default:  语句块n+1 //表达式和每个值比较都不等  } |

课后任务

(1)复习今天内容，整理思维导图

(2)练习：声明变量保存任意一个城市，使用switch-case判断，根据城市打印输出对应的美食;

练习：弹出两次提示框，分别输入商品的单价和数量，计算出商品的总价，假设总价满1000打9折，当前会员卡内有余额1200，如果足以支付，警示框弹出'pay success'，否则警示框弹出'pay error'

(3)预习循环语句

while do-while for