

day02 运算符及表达式及条件控制语句

运算符

算术运算符

- 在加法的时候如果出现字符串那么返回的绝对是一个字符串（会进行字符串拼接操作）
- 在其他相关的操作的时候会强制将其他的类型转为number类型再进行运算
- 如果在基础运算中没有出现字符串 而出现了NaN 那么结果必定是NaN (NaN不能进行算术运算)

++ 和 -- 前置后置的区别

- ++ 是自增（在原本的基础上加1 会改变原本的值） -- 是自减（在原本基础上减1 会改变原本的值）
- ++前置是先进行加1的操作 再执行后续的代码 ++后置先执行本行的代码再进行加1的操作

逻辑运算符

- && 全部都为true才是true 取得是最后一个true 只要一个为false那么对应的结果就是false 取得是第一个false
- || 只要有一个是true 结果就是true 那么取得是第一个true 全部为false结果才为false 取得是最后一个false
- ! 进行boolean类型的强制转换 取反将true变false false变为true
- 短路与 && 到达为false 后面的不执行 短路或 || 到达为true后面的不执行

比较运算符

- 返回的值都是boolean类型的值
- "1" == 1 比较值 "1"! == 1 比较值的同时比较类型
- NaN! =NaN
- 出现NaN大部分都是false (!=)
- 任意类型和数值进行比较会被转为数值
- 字符串和字符串比较那么比较的是ascii码（从第一位开始比较 大写A 65 小写a 97）

赋值运算符

- 赋值运算最后执行
- += -= *= ...都是会在原本的基础上进行赋值

位运算符

三目运算

boolean值?true的结果:false的结果

```
var a = 1+2>3?10:-10 //a的结果为-10
var i = 1?20:30 //20
```

表达式

表达式就是由运算符拼接的公式

- 算法运算符拼接的表达式 算术表达式

```
1+1+1
```

- 比较运算符拼接的表达式 条件表达式（转为boolean类型）

```
1+1>2
```

- 逻辑运算符拼接的表达式 关系表达式

```
1+1>2 && 1+2=3
```

运算符的执行顺序

运算符	描述
. [] ()	对象成员存取、数组下标、函数调用等
++ -- ~ ! delete new typeof void	一元运算符
* / %	乘法、除法、去模
+ - +	加法、减法、字符串连接
<< >> >>>	移位
< <= > >= instanceof	关系比较、检测类实例
== != === !==	恒等(全等)
&	位与
^	位异或
	位或
&&	逻辑与
	逻辑或
?:	三元条件
= x=	赋值、运算赋值
,	多重赋值、数组元素

示例

```
var n = 20
n++
var i = 10+29+(++n)-(n++)/10+30%7-5*3>10 && 20-15*29/18>13-(n++)/n+''+109 ? 18-
(2-4)*3+undefined-true+(n++) : n-(n++)*3+false+true //-47
console.log(n)//25
```

条件控制语句

概述：

程序控制语句所有的语言都具备，主要分为循环控制语句（循环执行）和条件控制语句（根据不同的条件执行不同的内容）

常用的条件控制语句

- if else （传递条件表达式 根据条件表达式来判断）
- switch case （空间复杂度大于if else 时间复杂度小于if else 传入一个值 根据来进行匹配）

if else

```
if(条件表达式){
    //条件满足执行的内容
}else{
    //条件不满足执行的内容
}
```

示例

判断一个数值是否奇数还是偶数

```
//prompt 弹窗输入框返回的是你的输入值 他的类型是string
var n = prompt('请输入你需要验证的数值')
if (n % 2 == 0) {
    console.log('是偶数')
} else {
    console.log('是奇数')
}
```

else if 多条件

```
//else if
//用于判断的方法 isNaN
if(isNaN(Number(n))){
    console.log('输入出错')
}else if (n % 2 == 0) {
    console.log('是偶数')
} else {
    console.log('是奇数')
}
```

if else 只会进入第一个找到的满足条件的内容

```
//if else 只会进入其他的第一个满足的条件 后续的不执行
if(2>1){
    console.log('进来了')
}else if(3>2){
    console.log('来玩玩')
}
```

示例2

输入的范围为100-999 判断一个数是否为水仙花数 （153 1的三次方+5的三次方+3的三次方 = 153）

```
var n = prompt('请输入100-999之间的数值')
if(n>=100 && n<=999){
    //if else嵌套
}
```

```

//取每一位的值
var a = n%10 //个数
var b = parseInt(n%100/10)//十位
var c = parseInt(n/100)//百位
if(a*a*a + b*b*b + c*c*c == n){
    console.log('是水仙花数')
}else{
    console.log('不是水仙花数')
}
}else{
    console.log('输入出错')
}

```

if else的嵌套

if else 允许多层嵌套 但是建议嵌套层级不要超过两层

```

if(条件表达式){
    //true的时候走的
    if(条件表达式){
        ....
    }else{
        //false的时候走的
    }
}else{
    //false的时候走的
}

```

示例

判断输入的数值的奇偶

```

var n = prompt('请输入你需要验证的数值')
//用于判断的方法 isNaN
if(!isNaN(Number(n))){
    if (n % 2 == 0) {
        console.log('是偶数')
    } else {
        console.log('是奇数')
    }
}else {
    console.log('输入出错')
}

```

输入一个数 判断是否为3的倍数 以及是否是15的倍数 (1-999)

```

// 输入一个数 判断是否为3的倍数 以及是否是15的倍数 (1-999)
var printIn = prompt('请输入1-999之间的数值')
if(printIn >= 1 && printIn <= 999){
    //取余3 取余15
    if(!(printIn % 3)){//值为0的情况下进入 整除3
        console.log('他是3的倍数')
        //是否为15的倍数
        if(printIn % 5 == 0){
            console.log('他是15的倍数')
        }
    }
}

```

```

    }else{
        console.log('不为3的倍数也不为15的倍数')
    }
}else{
    console.log('输入出错')
}

```

注意事项

- if else他只会进入其中的一个条件中 不会同时进入两个同级的代码块
- if 里面的条件可以为表达式也可以为值 但是都会被强制转换为boolean类型
- if else可以嵌套多层 一般建议嵌套不超过两层（可读性 可维护性）

switch case

```

switch(值){
    case 值1:
        执行的代码
        break
    case 值2:
        执行的代码
        break
    ...
    default:
        上面都不满足的默认的执行代码
}

```

示例

根据输入的指令打印对应的操作

```

/*
某个空调有对应的开关 当你按下的开关键为1号键的时候 执行加热操作
按下2号键 执行制冷操作
按下3号键 执行通风操作
按下的键为其他 不执行操作
*/
var code = prompt('请输入你需要执行的指令')
switch (Number(code)) {
    case 1://单独的分支
        console.log('正在执行加热操作')
        break //break 跳出 结束这个switch块 后续的不再执行
    case 2:
        console.log('正在执行制冷操作')
        break
    case 3:
        console.log('正在执行通风操作')
        break
    default:
        console.log('错误指令 不执行操作')
}

```

- switch 的值比对应用到的是=== 必须要类型和值都一致
- 每个case块都是一个分支 如果没有break那么会进入下一个分支
- 多个分支可以执行一个操作
- break用于跳出switch块 那么后续的分不再执行

- switch 不适用于范围内的比对
- default 默认的执行 上面都不满足的情况下执行


switch块允许多层嵌套

```
switch(n){
  case 值1:
    switch(i){
      case 值1:
        ...
        break
      }
    break
  case 值2:
    ....
  default:
    ...
}
```

示例

判断一个输入值的奇偶

```
var n = prompt('请输入值')
//判断是否输入正确
switch(isNaN(Number(n))) {
  case true:
    switch(n%2){
      case 0:
        console.log('偶数')
        break
      case 1:
        console.log('奇数')
        break
    }
    break
  case false:
    console.log('输入出错')
}
```

成绩表 输入对应的成绩 60分为及格 70分为一般 80分为良好 90 分为优秀 100分为  plus 其他分数输入无法判断

```
switch(prompt()-0){
  case 60:
    console.log('及格')
    break
  case 70:
    console.log('一般')
    break
  case 80:
    console.log('良好')
    break
  case 90:
    console.log('优秀')
    break
}
```

```

case 100:
    console.log('🐶plus')
    break
default:
    console.log('输入错误')
}

```

LOL 小学生之手 Q 大杀四方 W 致残打击 E 无情铁手 R 诺克萨斯断头台 A 鼠标点击 平A 其他键
defeat

```

switch(prompt()){
    case 'Q':
        console.log('大杀四方')
        break
    case 'W':
        console.log('致残打击')
        break
    case 'E':
        console.log('无情铁手')
        break
    case 'R':
        console.log('诺克萨斯断头台')
        break
    case 'A': case 'click':
        console.log('平A')
        break
    default:
        console.log('defeat')
}

```

- switch 适合实际的值的列举（枚举） if 使用范围内容条件判断
- switch 空间复杂度高于if switch时间复杂度低于if

☐ 枚举相当于一个箱子 箱子里面有很多对应的值 每个值有对应的名字

☐ 箱子 --- 1号 2号 通过列举的名字来获取里面的内容（里面内容为常量（不变的量））

- 课堂上所有代码至少写一遍, 熟练掌握今日所有知识点
- 判断一个整数, 属于哪个范围: 大于0; 小于0; 等于0
- 判断一个整数是偶数还是奇数, 并输出判断结果
- 开发一款软件, 根据公式 (身高-108) * 2 = 标准体重, 可以有10斤左右的浮动。来观察测试者体重是否合适
- 已知圆的半径 r , 求出圆的面积 s ($S = \pi \times r^2$)
- 随意输入一个年份, 判断这个年份是否为闰年.
- 输入赵本山的考试成绩, 显示所获奖励
 成绩 ≥ 100 分, 爸爸给他买辆车
 成绩 ≥ 90 分, 妈妈给他买MP4
 90分 $>$ 成绩 ≥ 60 分, 妈妈给他买本参考书
 成绩 < 60 分, 什么都不买
- 会员购物时, 根据积分的不同享受不同的折扣, 计算会员购物时所获得折扣

会员积分 x	折 扣
$x < 2000$	9折
$2000 \leq x < 4000$	8折
$4000 \leq x < 8000$	7折
$x \geq 8000$	6折

BMI指数 (即身体质量指数, 简称体质指数又称体重, 英文为Body Mass Index, 简称BMI), 是用体重公斤数除以身高米数平方得出的数字, 是目前国际上常用的衡量人体胖瘦程度以及是否健康的一个标准。主要用于统计用途, 当我们需要比较及分析一个人的体重对于不同高度的人所带来的健康影响时, BMI值是一个中立而可靠的指标。

体质指数 (BMI) = 体重 (kg) \div 身高² (m)
 如: $70\text{kg} \div (1.75 \times 1.75) = 22.86$

成人的BMI数值:
 过轻: 低于18.5
 正常: 18.5-24.99
 过重: 25-28
 肥胖: 28-32
 非常肥胖, 高于32

示例:

按照考试成绩的等级, 输出百分制数段.

