NaN(not a number):

它是一种数字的特殊情况，也是number类型，是计算发生错误的一种状态。

判断NaN的情况：isNaN()

自增，自减运算：

自增是在数据原有的基础上加一。

自减是在数据原有的基础上减一。

先自增和后自增的区别？

如var b = a++ +10; 和 var b=++a + 10;的区别是什么？

与或非混合运算时，该怎么计算？

如： !true && true || false ,该怎么计算得到最终的结果

应该按照先非运算，再与运算，最后或运算的顺序进行计算。

显示数据类型转换：

A.转换为数字类型：

1. Number():如果遇到不能转换的内容，结果会是NaN。空字符串转换为0
2. parseInt():忽略字符串前面的空格，直到遇到第一个字符开始进行转换，一直转换到遇到无法转换的字符为止。
3. parseFloat():与parseInt()类似，parseInt转换的结果是整数，parseFloat转换的结果是小数

B.转换为字符串类型：

1. String():将其他类型的数据转换为字符串类型
2. toString():将其他类型的数据转换为字符串类型

C.转换为boolean数据类型：

1. Boolean():将其他类型的数据转换为boolean类型

注意：只有NaN,””,0,”false”,undefined转换的结果为false，其他内容转换的结果都是true。

隐式数据类型转换：

1. 转换为数字类型：除了加法以外，进行其他的运算都会将数据隐式转换为数字类型
2. 转换为字符串类型：将数据和空字符串用加法连接，就会将数据隐式转换为字符串类型。
3. 转换为boolean类型：！！数据，将数据取反再取反，就会将数据隐式转换为boolean类型。

比较运算符：

1. > 大于
2. < 小于
3. >= 大于等于
4. <= 小于等于
5. == 相等
6. != 不相等
7. === 全相等（内容一致，数据类型也要一致）
8. !== 不全相等（内容不一致或者数据类型不一致）

if判断语句：

1. if...else语句语法格式：

//当条件为true时执行代码段1，条件为false时执行else中的代码段2

if(条件)

{

代码段1;

}

else

{

代码段2;

}

1. if...else if...else 语句语法格式:

//只要有一个条件成立，则执行对应的代码段，如果没有一个条件成立，则执行代码段else。

if(条件1)

{

代码段1;

}

else if(条件2)

{

代码段2;

}

......

else if(条件n)

{

代码段n;

}

else

{

代码段else;

}

1. if 语句语法格式：

//如果条件成立，则执行代码段1，如果条件不成立，则继续执行if语句后面的其他代码

if(条件)

{

代码段1;

}

断点调试：使用开发人员工具来调试编写的js代码，每一个合格的前端开发都必须熟练使用这种方式。

switch语句:

用来判断可能出现的多种情况。

语法格式：

//如果需要判断的内容是情况1则执行代码段1，如果需要判断的内容是情况2则执行代码段2。。。。如果需要判断的内容是情况n则执行代码段n。如果从情况1~情况n都不符合则执行default对应的代码段default。

switch(需要判断的内容)

{

case 情况1: 代码段1; break;

case 情况2: 代码段2; break;

.......

case 情况n: 代码段n; break;

default: 代码段default; break;

}

三元运算符：? :

三元运算符的语法格式：

boolean表达式?代码段1:代码段2;

判断boolean表达式的结果如果为true则执行代码段1，如果boolean表达式的结果如果为false则执行代码段2.