数据类型：原始类型和引用类型。

原始类型：

数值型：（var a = 4;//默认10进制,var b = 010;//8进制以0开头。16进制，以0x开头，不区分大小写。10-15，a-f。var c =0xa/var c =0xA）

整形：var n = 3.1415e2(3.1415\*10^2)/3.14e-2<====>3.14\*(10^-2)

浮点型：var n = 314.15

typeof:检测数据类型。

typeof 数据 检测数据类型。typeof n

字符串型：

用引号所包含的数据。

查看任意一个字符的编码。charCodeAt()

布尔型：

true、false 常用于记录状态、是否注册、是否为会员、运算结果。

未定义型：undefined

声明一个变量，未赋值，

空：null

只有一个值null，类型是object，常用于和对象（引用类型）数据一起使用。

数据类型的转换：

隐式转换：运算过程中自动发生的转换，不需要人为干预。

1、数值+字符串：数值会转换为字符串，1+'2' <=====> 12(string)

2、数值+布尔型：布尔型会转换为0或1,1+true <=====> 2(number)

3、布尔型+字符串：布尔型=>true或false，true+'hello' <=====> truehello(string)

4、var m5 = '5' - '3a' //NaN not a number

5、var m6 = 1 + null //1，null=>0

6、var m7 = 1 + undefined; //NaN undefined=>NaN.

- 、\* 、/ 都会转换为number

加号 + 作用：

1、运算符（+），数值间加法运算。

2、拼接字符。

隐式转换特点（加减乘除）：

加号+两端有一个字符串，则另一个会转换为字符串。

其他情况（减乘除）会将非数字隐式转换为数值型，转换过程中自动调用函数Number()来完成的。

NaN和任何数字执行加减乘除操作都会得到NaN

强制转换：

1、强制转换为数值型。

// 强制转为数值型

var n = Number('5'); //5

var n1 = Number('5a'); //NaN

// console.log(n1,typeof n1);

var n3 = Number(true) //1、0

// console.log(n3,typeof n3)

var n4 = Number(null) //0

var n5 = Number(undefined) //NaN

console.log(n4,n5)

NaN:not a number不是一个数字，没有成功转为数值返回这个结果，NaN和任何数字执行加减乘除运算即结果还是NaN。

2、强制转为整形。

通常用于将字符串和小数转为整形。

parseInt()

parseInt(3.14) ===>3

parseInt(-3.14)===>-3

parseInt('6.18a')====>6

parseInt('a6.18b') ====>NaN

转为整形的时候，会从数据的开头寻找整数，找不到返回NaN。

3、强制转为浮点型。常用于将字符串转为浮点型。

var a1 = parseFloat(3.14); //3.14

var a2 = parseFloat('6.18a');//6.18

var a3 = parseFloat('a6.18b');//NaN

var a4 = parseFloat('5a');//5

4、只能将数值和布尔型转为字符串。toString()

var b1 = 2;

var new\_b1 = b1.toString();

console.log(new\_b1,typeof new\_b1) //2,string

4、运算符。

表达式：由数据或由运算符连接的数据所组成的形式。

运算符包含6类，算术运算符、比较运算符、逻辑运算符、位运算符、赋值运算符、三目运算符。

①算术运算符：+、-、\*、/、%（取余）、++、--（具有隐式转换为数值）

console.log('5'%'3'); //2

如果单独执行自增（++）或自减（--）++a和a++两者没有区别。

var a =1 ;

var b = a++; //先把a的值赋给b，再执行a自增。

var c = 3;

var d = --c; //先执行c自减，然后在把自减后的c赋值给d

var a = 6;

var num1 = a--; //1、先将a赋给num1,num1=6,2、a自减，a=5

var num2 = --a; //2、先自减，在把自减后的值赋值给num2。

console.log(num1+num2)

等价于：

console.log(a-- + --a);

②比较运算符。

==：只比较值是否相同，可能发生数据类型的转换。

===：比较值和类型。（推荐使用）

！=：只比较值。

!==：先比较类型，后比较值。（推荐使用）

// 练习：3和'10'比较

console.log(3>'10'); //false;字符串隐式转换为数值。

console.log('3'>'10'); //true

// 字符串比较，会比较首个字符的Unicode码

console.log('3'.charCodeAt(),'1'.charCodeAt())

'10a' ==>NaN，NaN和任何值进行比较都是false，（> < == >= <= ===）NaN==NaN也是false。

console.log(3>'10a'); //false

console.log(3<'10a'); //false

console.log(3=='10a') //false

③逻辑运算符。

&&（逻辑与。两个条件都是true结果才是true，否则是false）

||（逻辑或。关键的两个条件只有有一个是true结果就是true，否则是false）

!（逻辑非。取反，）

短路逻辑：

&&：当第一个条件为false，不再执行第二个条件。

|| ：当第一个条件为true，不再执行第二个条件。

④位运算符。

先转2进制，再把2进制转为10进制。

& 按位与，上下两位都是1，结果是1，否则是0.

3&5

011

101

001(2进制)==>1(10进制)

5&7

101

111

101(2进制)===>5(10进制1\*(2^2)+0\*(2^1)+1\*(2^0))

| 按位或，上下两位含有1就是1.

5 | 7

101

111

111（2）==>(7)

6 | 9

0110

1001

1111(2)===>15

⑤赋值运算符。

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

运算赋值，先运算后赋值。

⑥三目运算符（三元运算符）。

一目运算符：由一个运算符连接的一个操作数据。（++、--、!）

二目运算符：由一个运算符连接的两个操作数据或者表达式。

三目运算符：由两个运算符连接的三个操作数或表达式。(sex==1?'男':'女')

浏览器端函数：

alert：消息框/警示框

prompt：弹出提示（输入框），通过变量保存用户输入的数据。得到字符串类型数据，不输入内容，得到一个空字符串 '',取消得到null object.

程序：数据 + 算法

程序执行方式：顺序执行、选择执行、循环执行。

流程控制（选择执行）

1、if语句

if(3){ //true

}

if(''){ //false

}

隐式转换为数值型

如果有书库作为条件表达式，会隐式转换为布尔型

false:0、NaN、''(空字符)、undefined、null ===>false

var n = 1 || 2;

//判断第一个是否为true，如果是true则把第一个数字赋值给n，如果是false，则把第二个数字赋值给n

var m = 1 && 2;

判一断第个数字是否为true，如果是true执行第二个，将第二个数字赋值给m，如果是false，不再执行第二个，将第一个数字赋值给m。

2、if-else语句

if(条件表达式){

}else{

}

3、if-else嵌套。

if(){

}else if(){

}else{

}

4、switch-case（switch中表达式的值与case中的值比较时，只能使用===，要求值和类型都相同）

是一种特殊的多项分支语句，根据一个值选择执行相应的语句。

switch(表达式){

case 值1: //用表达式和值1进行比较，如果相同则会执行后边的语句。

语句块1;

break; //结束switch，不再往后执行。

case 值2:

语句块2;

break;

default: //如果表达式和以上每个值比较都是false，才会执行default。

语句块 n+1;

}

复习。

思维导图。

声明变量保存年份，如果是闰年打印闰年，

能被4整除不能被100整除，或者能被400整除。

((year % 4 == 0 && year % 100 != 0) || year % 400 == 0) && console.log('闰年')