1数据类型

分为原始类型和引用类型

原始类型分为数值型，字符串型，布尔型，未定义型(undefined),

空（null）

1. 数值型

分为整型和浮点型

整型在内存中占4个字节，浮点型占8个字节，

8进制，以0开头，例如012->10

16进制，以0X开头，例如0XF->15

1. F 代表10~15 不区分大小写

0XFF->255

浮点型分为普通小数和指数小数

1234.56 ->1.23456e3 -> 1.12345\*10^3

0.0000034 -> 3.4e-6 -> 3.4\*10^-6

typeof(数据) 检测数据的类型，

1. 字符串型

数据被引号所包含就是字符串类型。不区分单双引号。

查看某一个字符的Unicode码，

65533 说明了没有修改utf8

‘华’.charCodeAt(); //返回一个数字

1. 布尔型

在程序中表示真或者假的结果

true/false

常用于一些是否的结果，例如：是否为会员，是否注册。是否在线

Is Member=true is Online=false

练习：检测true 或者false 的数据类型

1. 未定义型

声明了变量未赋值，结果就是未定义型。只有一个值undefined

1. 空

常用于和引用类型一起使用，只有一个值null

typeof(数据) 检测数据的类型，

数值型->number

字符串型->string

布尔型->boolean

未定义型->undefined

空-> object

1. 数据类型转换
2. 隐式转换

1数值型+字符串型：数值型被转成了字符串类型

1+’2’ //’12’

2数值型+布尔型：布尔型被转成了数值型true->1 false->0

3+true //4

3+false //3

3布尔型+字符串型:布尔型被转成字符串型

True+’hello’ //’truehello’ //‘5’+true //5true

JS中加号（+）的作用

执行加法运算;执行字符串的拼接

练习：查看一下程序的执行结果

var a=3,b=true,c=’tedu’;

console.log(a+b+c);// 4tedu

console.log(b+c+a);//truetedu3

console.log(c+a+b);//tedu3true

var num='3'-1;

var num1=2-null;

var num2=3-'a';

var num3=1-undefined;

console.log(num, typeof num);//2 number

console.log(num1,typeof num1);//2 number

console.log(num2,typeof num2);//NaN number

console.log(num3,typeof num3);//NaN number

隐式将‘a’转为数值型，返回NaN //(Not a Number) 不是一个数字，任何值和NaN执行加减乘除运算符都会返回NaN

(2)强制转换

1将任意数据转成数值型

Number()

|  |
| --- |
| Number(a); //NaN  Number(‘1’)//1  Number(‘1a’)//NaN  Number(undefined)//NaN  Number(null) //0  Number(true)//true->1 false->0 |

练习：计算’3’-1, ‘2a’/5 ,true-null的结果，并查看数据类型。

var p='3'-1;

console.log(p);

console.log(typeof(p)); //2 number

var e=Number('2a'/5);

console.log(e);

console.log(typeof(e)); //NaN number

var f=Number(true-null);

console.log(f);

console.log(typeof(f)); //1 number

结论：在执行减法，乘法，除法运算的时候，如果数据不是数值型，则自动调用Number转为数值型

2.将任意的数据转为整型

Parselnt(数据)

Parselnt(“4.5a”)//4

Parselnt(‘a4’) //NaN

常用于将字符串或数字转为整型。其它的数据返回NaN,如果要转换的字符串以非数字开头，也返回NaN

转换规则；如果要转的数据 是字符串，从开头开始查找数字，遇到小数或者非数字，则返回前边的数字，否则开头不是数字 ，返回NaN ,如果是布尔型，未定义型，空返回NaN.

3 将任意数据 转为浮点型

parseFloat(‘4.5a’); //4.5

转换规则：和parselnt 用法基本上一致，只是转的数据是浮点型。

‘4 将数值型和布尔型转为字符串型

|  |
| --- |
| var num=10;  var str=num.toString();  console.log(str,typeof str);// |

toString()

Var a=2;

1. toString(); //’2’
2. toString(16);如果要转的数据是数值，可以设置进制（2.8.16）
3. 运算符

由运算符连接的操作数据,所组成的结构称为表达式

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

++ 自增， 在原来的基础之上加1

-- 自减。 在原来的基础之上减1

console.log(a++) ,先打印a的值 ，再执行自增

console.log(++a); 先执行自增，再打印a的值

练习 ：计算以下代码的执行结果

var num=3;

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

//先取出num的值3，然后执行自减，num的值变成了2

//先执行自减，num的值变成了1，然后再取出num的值，也就是1

Var b=3;

console.log(b++ + ++b);

1. 比较运算符

> < >= <= ==(等于) !=(不等于) ===(全等于) !==(不全等于)

==只是比较两个数据的值是否相同

==== 不仅比较值，还会比较类型是否相等，都相同返回true

|  |
| --- |
| 如果数据类型不同，会发生数据类型转换 |

1. ’10’ //字符串转成数字10 //false

数字和字符串比较，字符串转成数值型

‘3’>’10’ //true

两个字符串比较，比较的是Unicode码

1. >51 1->49

‘张无忌’>’张翠山’ //false

如果首个字符相同，会继续往后比

3 >’10a’ //false

3<’10a’

3==’10a’

数值和字符串比较，字符串自动调用Number转为数值型。Nunmber(‘10a’) ->NaN

NaN和任何值比较 （> < <= >= == === ），包括自身，返回false

NaN==NaN //false

1. 逻辑运算符

&& 并且，关联的两个条件都是true,结果才是true,否则false

|| 或者，关联的两个条件只要一个为true ,结果就是true,否则false

! 取反 !true ->false !false->true console.log( !(3>1) );

练习：声明两个变量保存用户名和密码，如果用户名是’root’,并且密码是’123456’,打印true,否则打印false

var uname='root';

var upwd='123456';

console.log(uname=='root' && upwd=='123456');

练习：声明一个变量保存年龄，如果年龄90岁以上，或者3岁以下，打印true,否则打印false

逻辑短路

&& 当第一个条件为false,就不需要继续执行第二个条件

|| 当第一个条件为true,就不需要继续执行第二个条件

逻辑短路无需关注最终结果是true还是false，重点是看是是否会执行第二个表达式

练习：以下两语句是否会报错

var num=3;

num>5&& console.log(a); //不会 ,./.

num<1|| console.log(a); //会

声明变量保存一个人的年龄，如果满18，岁，打印'成年人';

(4)位运算符(了解)

在执行运算的时候，把数字转成二进制再进行运算

1 10 11 100 101 110 111 1000 1001 1010

2 4 6 8

将11010用10进制表示？

11010=10000+1000+10 ->26 一个0一个1次 3个0代表2的3次

16 + 8 + 2

101101=100000+1000+100+1 ->45

32 + 8 + 4 +1

将25用二进制表示？

16 + 8 +1

10000+ 1000+1=11001

按位与(&) 上下两位都是1，结果是1，否则是0

按位或(|) 上下两位含有1，结果是1， 否则是0

按位异或(^) 上下两位不同为1，相同为0

4^7 5^8

/\*100 0101

111 1000

011 1101

\*/

按位右移(>>) 删除二进制的最后的1位或者,每次缩小到原来的一半或者更多

按位左移(<<) 在二进制的最后添加个0，成倍增加

课后任务

1. 复习今天的内容，整理思维导图
2. 练习

声明一个变量保存年份，判断这个年份是否为闰年，如果是打印‘闰年’。

闰年：查看闰年的判断条件（能被4整除，并且不能被100整除，或者能被400整除）

(3)预习if,if-else,if-else 嵌套，switch语句