一.常量

常量用来存储数据的容器

常量声明后必须赋值，不允许重新赋值

const a=1;

使用const关键字声明常量

二.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、空

1.数值型

10进制

1 2 3 .. 7 8 9 10 .. 15 16

8进制：以0开头的数字，例如：012

1 2 3 .. 7 10 11 12

16进制：以0X开头的数字，不区分大小写，例如：0XA

1 2 3 .. 7 8 9 a ~ f 10

浮点型

3141.5e-1

314.15

31.415e1

3.1415e2

|  |
| --- |
| typeof 检测数据类型  number/string/boolean/undefined/object |

2.字符串型

所有被引号包裹的数据就是字符串型

查看任意一个字符的Unicode码

'a'.charCodeAt()

3.布尔型

只有两个值，分别是true和false，代表真和假

用于保存只有两个值的数据，例如：是否为会员、是否登录、是否已婚...

4.未定义型

只有一个值undefined，声明变量未赋值则值为undefined

表示空值

5.空

只有一个值null，常用于对象中

三.数据类型转换

分为隐式转换和强制转换

1.隐式转换

在运算过程中自动发生的转换

(1)数字+字符串 数字转为字符串

2+'3' //'23'

(2)数字+布尔型 布尔型转为数字，true->1 false->0

2+true //3

2+false //2

(3)字符串+布尔型 布尔型转为字符串

'3'+true //'3true'

|  |
| --- |
| **加号(+)的作用**  数值之间的加法运算  字符串之间的拼接 |

练习：查看以下程序的输出结果

var a=1,b=true,c='tedu';

console.log(a+b+c);//'2tedu'

console.log(b+c+a);//'truetedu1'

console.log(c+a+b);//'tedu1true'

|  |
| --- |
| NaN：Not a Number，不是一个数字，将一个值转为数字的时候，没有成功得到数字结果就是NaN  NaN和任何数字执行数学运算，结果还是NaN |

所有的隐式转换为数字自动调用函数Number

2.强制转换

(1)强制转为数字

Number()

|  |
| --- |
| Number('1') //1  Number('1a') //NaN  Number(true) //1  Number(false) //0  Number(undefined) //NaN  Number(null) //0 |

(2)强制转整型

parseInt()

用于将字符串和小数点位整型，其它转换的结果为NaN

|  |
| --- |
| parseInt(1.9) //1  parseInt('3.14') //3  parseInt('6a6') //6  parseInt('a66') //NaN |

(3)强制转换为浮点型

parseFloat()

用于将字符串强制转换为浮点型

|  |
| --- |
| parseFloat('3.14') //3.14  parseFloat('6a') //6  parseFloat('a6.18') //NaN |

(4)强制转换为字符串(了解)

toString()

用于将数值和布尔型转为字符串

|  |
| --- |
| var n=5;  var str=n.toString(); //'5' |

四.运算符

表达式：由数据本身或者由运算符连接的操作数据组成的形式

运算符分为算术运算符、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，让一个变量在原来基础之上加1

-- 自减，让一个变量在原来基础之上减1

|  |
| --- |
| var a2=5;  //先把当前a2的值赋给a3，然后a2执行自增  var a3=a2++; |
| var a4=5;  //a4先执行自增，然后再把自增的结果赋给a5  var a5=++a4; |

练习：查看以下程序的运行结果

var n1=3;

var n2=n1--;

var n3=--n1;

console.log(n1,n2,n3);

2.比较运算符

> < >= <= ==(等于) != ===(全等于) !==

== 等于，只是比较值是否相同

=== 全等于，比较值和类型是否相同

!= 不等于，只是比较值

!== 不全等于，比较的值或者类型

(1)字符串和数字比较，字符串转为数字

10>'3' //true

(2)两个字符串比较，比较的首个字符的编码

'10'>'3'

(3)查看以下程序运行，哪个是true

'10a'>3 //false

'10a'<3 //false

'10a'==3 //false

NaN==NaN //false

NaN和任何值比较(> < >= <= == ===)结果都是false

3.逻辑运算符

&& 逻辑与，关联的两个条件都是true结果是true

|| 逻辑或，关联的两个条件有一个是true结果是true

! 逻辑非，取反

短路逻辑：不再执行第二个条件就是发生了短路

&& 当第一个条件为false的时候，不再执行第二个条件

|| 当第一个条件为true的时候，不再执行第二个条件

|  |
| --- |
| 练习：查看以下程序运行是否报错  var a=2;  a>1 || console.log(num);  a<5 && console.log(num); |

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年份，判断是否为闰年，如果满足打印true，否则打印false

（闰年有3个条件）

(3)预习js第3天

http://www.codece.com/?p=193