正课:

1. 数据类型转换：

2. \*\*运算符和表达式:

1. 数据类型转换:

弱类型语言:

1. 声明变量时不必提前指定数据类型

2. 同一个变量先后可保存不同类型的数据

3. 程序会根据自身的需要自动改变数据的类型

什么是数据类型转换: 将不需要的数据类型转化为需要的数据类型

为什么: 有时，变量中获得的数据类型不是想要的。

何时: 只要获得的数据类型不是想要的，都要转换

如何: 2种:

1. 隐式转换:

什么是: 程序根据自身的需要，自动将不符合要求的数据类型转为需要的数据类型。不用程序员干预.

何时: 只要所给的数据类型不是程序想要的

如何: 每种运算中都自带隐式转换

只看算术计算中的隐式转换:

规则: 1. 默认: 一切数据都转为number类型再计算

bool: true -> 1 false -> 0

2. 特殊: +运算中,如果碰到字符串，则不再转数字，而是都转字符串，+运算变为字符串拼接

2. 强制转换: 程序员主动调用功能，完成的转换

何时: 只要隐式转换的结果也不是想要的

如何: 3种:

1. 转数字: 2种:

1. 将纯数字或bool转为number:

var n=Number(x);

强调: 不直接修改原变量中的值，而是返回新值。

问题: 只能转纯数字组成的字符串和bool类型

如果转不了: 返回NaN:

什么是: NaN(Not a Number)

代表一切不是数字的值

何时: 专门用于判断类型转换或算术计算错误。

特点: NaN参与任何计算，结果只能是NaN

2. 将任意字符串转数字:

var n=parseFloat(str);

强调:1. 可去掉结尾的非数字字符

但无法去掉开头的非数字字符

2. 如果传入的值不是字符串，则先隐式转为字符串，再尝试转为数字。

比如: Number(true) -> 1

parseFloat(true) -> parseFloat("true")

-> NaN

var n=parseInt(str);

用法: 同parseFloat

强调: parseInt可去掉小数部分

如果转不了: 返回NaN

总结: 只要强制转为Number，都首选parseFloat

只有明确要去掉小数时才用parseInt

Number专门用于隐式转换中自动调用

特殊: Number("") -> 0

Number(null) -> 0

Number(undefined) -> NaN

强调： Number/parseFloat/parseInt都能去掉字符串中的空格后，再转数字。

2. 转字符串:2种:

var str=x.toString(); 不能转null和undefined

var str=String(x); 万能

总结: 首选String(x);

多数情况，由程序在隐式转为字符串时自动调用。

3. 转布尔: Boolean(x)

何时: 都是在隐式转为bool时，自动调用

规则: 只有5个值会被转 为false:

"" 0 null undefined NaN

其余都转为true

练习: 凡是从页面上获得的都是字符串

2. \*\*运算符和表达式:

什么是程序: 人的想法在计算机中的执行

什么是运算符: 程序中模拟人的想法的符号

什么是表达式: 由数据，变量和运算符组成的执行一项专门任务的程序语句

算术运算: + - \* / %

%模运算/取余数: 被除数/除数, 不要商，要除不尽的余数部分

何时: 2种:

1. 取零头, 可限制一个结果不超过指定最大值

2. 判断能否整除:

偶数: n%2==0 说明n是偶数

奇数: n%2==1 说明n是奇数

判断润年:

隐式转换:

默认: 都转为number,再计算

特殊: +运算中，如果碰到字符串，就都转为字符串，再拼接