**Js核心语法 day02**

**内置方法：**

**toString（）**

**Number（）**

**Boolean()**

**parseInt()**

**parseFloat()**

**prompt()**

**typeOf()**

**document.writer()**

**目标:**

1. 表达式
2. 运算符---重点
3. 函数---重点 难点

运算符的分类

1. 单目运算符 b++
2. 双目运算符 a+b a>b
3. 三目运算符 a?b:c

算数运算符

+ - \* /

% 10%3=1

++: i++ 先取值然后++ ++i :先加加在取值

‘—’:任意类型 ‘—’都会自动转换数字类型

如果有参数，不能自动转为数字，则为NaN

关系运算符

> number和string做比较的时候，自动会把string转换成number

<

>=

<=

!=

==

===

!==

= === ===的区别？

=：赋值的意思

==:只是值来比较（自动类型的转换）

===：值要比较，类型也要比较（不存在自动类型转换）

练习题：

编写一个程序，输入三位数的整数，输出时反向输出

**逻辑运算**

与 或 非

&&

|| 或者

! 取反的意思

**三目（三元）运算符**

语法： 表达式1？表达式2：表达式3

要求：表达式1必须是boolean，表达式2和表达式3无所谓

含义:表达式1为true的时候，则执行表达式2

表达式1为false的时候，则执行表达式3

有多个条件： 条件1？条件1的值:

条件2？条件2的值:

条件3？条件3的值:

......

默认值；

练习题：

1. 判断偶数还是奇数，要求有弹出提示框
2. 会员卡四位数字求和，大于20则返利50，不然不反
3. Var a=123; var b=234; 要求交换a和b的值
4. 判断输入的年份是不是闰年（prompt）

函数（方法）

概念：专门执行一系列专门任务的代码块

声明式的创建函数：有提升的功能

定义: function 函数的名称（形参列表）{

//执行的步骤

return 值;

}

调用: 函数的名称(实参的列表)

return

第一种方式：只要单独的一个return的时候，后面的语句不会执行（退出的功能）

第二种方式：return+值 有退出又有返回值得意思

为什么要有参数？

更加的灵活，参数是在方法的内部操作

形参：方法内部的独有变量

需求：写一个方法，这个方法专门用来计算两个数字的和，和要返回，并且在控制台输出。

**作业:**

1. 把本地的项目通过git上传到github
2. 团队协作
3. Vs code 可以直接用git的命令（研究一下）

Svn：集中式的

Git：分布式的

**作用域**

1个变量的可用的范围

全局的作用域：是放在全局的作用域中的变量

局部的作用域： 声明在函数的变量或者方法的参数

变量：全局的变量、局部的变量

变量的声明周期：

全局的变量周期:打开这个网页，全局变量就会创建，一旦你关闭网页就会销毁

局部的变量的声明周期：方法调用的时候创建，方法调用完成之后销毁

提升：1.变量有提升

1. 声明式的创建函数：有提升的功能

作业：

1.页面中弹出一个提升框，接受用户输入的年份，控制台输出这个年份是不是闰年（要求用方法）

**2.**页面中弹出一个提升框，读取文本的信息，判断这个文本是不是英文字符，是不是汉字，是不是一个数字（要求用方法）