**js day02**

**js的特点：基于对象 纯文本 解释执行 弱类型**

**弱类型：可以随意的更改类型，没有指定数据类型，通过值就可以判断是什么类型。**

**什么变量？从内存的角度来看**

**存储数据的空间**

**基本数据类型存储在栈里面**

**引用数据类型存储在堆里面**

**数据类型有哪些？**

**原始数据类型和引用数据类型**

**原始数据类型：number boolean undefined null string symbol（es6）**

**const a=8;**

1. **数据类型转换**

隐式转换：程序自动转换数据类型

强制转换：通过js提供的内置的方法去改变数据类型（自己说了算）

隐式转换：

如果只考虑+的情况下：

1. 只要有字符串参与，一切都变成字符串（相当于是字符串的拼接）
2. 如果没有字符串参与，都转为数字进行计算（true--1 false--0）

强制转换：通过js提供的内置的方法去改变数据类型（自己说了算）

概念：调用了专门的方法手动的转换类型（系统内置的方法）

1. x.toString() 转换为string类型

其中x可以是任意的类型（任意的类型都有toString的方法）

1. Number(x):将任意的x的类型转换成number类型

问题：

js一定放到script标签里面

typeOf():检测数据类型

console.log()

每结束一条语句，最好加;

1. parseInt()转换成整数

var s=parseInt(str):将字符串转换整数，只读取字符串整数的部分

1. 从第一个字符往后面读
2. 如果碰到一个数字字符，开始读取，再次碰到不是数字的字符，则停止读取。
3. 如果开头碰到空格，可以继续往后读取。
4. 如果碰到一个非空格的字符，不是数字说明不能转--NaN

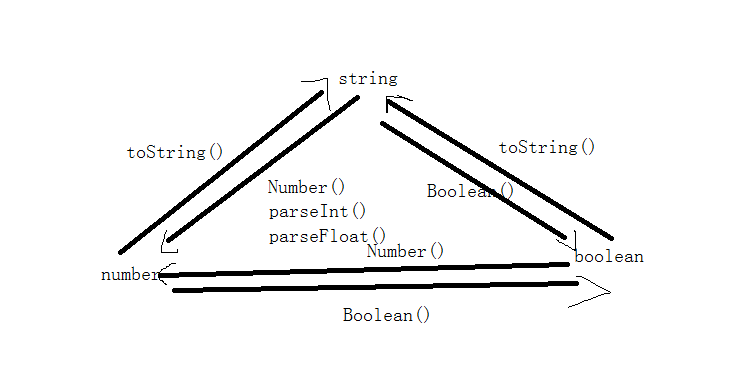
NaN：不是数字（内容）的数字（类型）

4.parseFloat(“str”):转为浮点数

用法和上面parseInt()差不多

唯一的区别：parseFloat认识小数点，但是它仅仅只认识第一个小数点

数据类型转换的图



总结一下：到目前为止，现在学了哪些内置的方法

toString() Number() parseInt() parseFloat() Boolean() typeof()

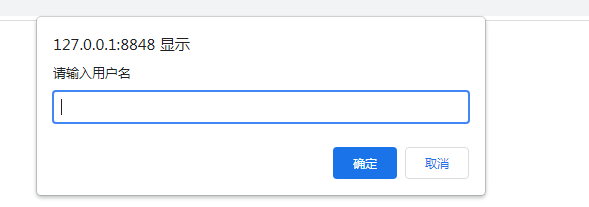
console.log() alert()

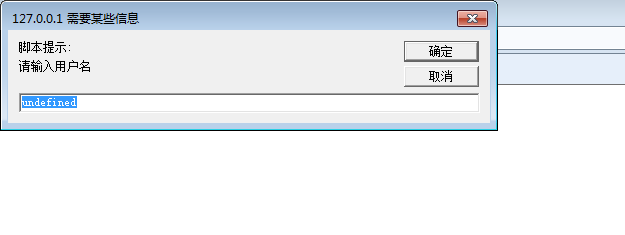
onclick事件 键盘事件 鼠标事件

bom：没有标准的

prompt(“提示的信息”):专门用来请求用户输入数据的对话框

1. 不好看 不好改 每个浏览器的样式不一样





练习题（数据类型的转换）

1. 弹出一个输入提示框，请用户输入一行数字，控制台输出该输入数据的数据类型。
2. 尝试用户的输入数据+1 输出到控制台里面。
3. 尝试把用户的数据转换成一个整数，该整数加1,控制台输出结果。
4. 尝试用户数据的数据转换为浮点数，把浮点数+0.01输出到控制台里面。

prompt(“提示的信息”)

得出一个结论：prompt()输出的结果是字符串的类型

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

算数运算符

+

-

考虑-：任意类型- ，都会自动转换为数字类型

如果有参数参与，就不能自动转换为数字，返回NaN

\*

/

%

++:i++先取值后+ ++i 先加再取值

--

关系运算符

> number类型和string类型做比较的时候 自动把string类型转换number类型

<

>=

<=

==:值来比较（自动类型转换）

===：值比较并且他们的类型也要比较（不存在自动类型的转换）

！=

= == ===的区别？

=赋值

==：只比较值

===：要比较值也要比较类型（绝对等于）

1. 函数