js笔记 2019/04/09 20:37

（一种解释性语言，逐条执行）

基础知识

~~javascript语句打不打分号都行，因为会自动补分号，但是为了代码稳定，还是打分号吧~~

呸 就得打分号

js中“==”与“===”的区别：

==用于一般比较===用于严格比较

首先，== equality 等同，=== identity 恒等。

==， 两边值类型不同的时候，要先进行类型转换，再比较。

===，不做类型转换，类型不同的一定不等。

绝对等于就是要一模一样，除了NaN === NaN为false NaN == NaN也为false

一言以蔽之：==先转换类型再比较，===先判断类型，如果不是同一类型直接为false。

undefined == null 但是 undefined === null是错误的

{} == {} --》false引用值比的是地址，相当于两个房间虽然一样但是地址不同

javascript 尽量避免小数运算，js自带精度不准的bug：

eg ： 0.14 * 100 = 14.00000000000002 可能产生类似的这种尤其是0.14必然会出错

js只能操作小数点后十六位，小数点前十六位。

可用：

Math.floor(num) 向下取整

Math.ceil(num)向上取整

Math.random()产生一个随机数（0~1）之间

n.toFixed(num)保留num位小数

1. js中单引号和双引号都代表字符串，但是为了和后端语言相配合（PHP没有双引号），最好写单引号
2. document.write(num.toFixed(1)) 打印保留几位小数（）内为位数
3. window.prompt('input') 输入语法
4. if 里面不能定义function，以前可以
5. str.charAt() 把字符串看成列表形式，括号内填下标，可以打印出相应的字符
6. str.charCodeAt () 可以返回指定位置字符的Unicode，这个返回值是0 - 65535之间的整数如果Unicode编码大于255证明是两个字节，小于255是一个字节

7. typeof()　　返回括号内类型，加不加括号都行

可以输出没定义变量类型，结果为undefined　　string object 不能返回null算object

8. typeof(null)结果为object对象属性

9. isNaN() 函数用于检查其参数是否是非数字值。要经历Number()转换完再判断

10. 预编译中this指向Window

11. 字符串"10">"9"是错的字符串比是比较Ascii码，如果是"10">9是正确的因为字符串会转化为数字

12. var num = 234这种也叫做window的属性，叫不可配置的属性 经过var的过程delete不掉，形参也同样删除不掉

13. indexOf()　索引，返回数组中某个元素的位置　返回数字，如果没找到则返回-1

parseInt() 函数可解析一个字符串，并返回一个整数。

parseInt(*string*, *radix*)

其实就是转换数字进制 radix表示几进制

说明

当参数 *radix* 的值为 0，或没有设置该参数时，如果 *string* 以 **"0x"** 开头，parseInt() 会把 *string* 的其余部分解析为**十六进制**的整数。如果 *string* 以 **0** 开头，那么 ECMAScript v3 允许 parseInt() 的一个实现把其后的字符解析为**八进制或十六进制**的数字。如果 *string* 以 **1 ~ 9** 的数字开头，parseInt() 将把它解析为**十进制**的整数。

NaN　not a number a 表示非数

alert(num)　　浏览器提示框输出

console.log(num)　　在控制台输出

```
function sum(a,b)
{
        函数体
        形参和实参可以不对等
        如果实参大于形参，所有实参都会存入arguments　　argument是数组形式的实参列表
        sum.length表示形参长度
        a变，argument[0]也变，argument[0]变，a也变，但a与argument不是同一个变量，内部的映射规则导致必须一样
}
```

函数声明整体提升，变量 声明提升

函数声明能提升，表达式不能提升 var b = function () {} 这不属于函数声明

“也就是说明JavaScript在函数调用时，函数写在调用前后都可在预编译阶段就已经声明过函数变量”

window（全局对象）就是全局的域，window像一个仓库，所有变量都能通过window.a调用出来

1，任何变量，如果变量未经声明就赋值，此变量为全局对象所有。

2，一切声明的全局变量，全是window的属性

eg:

```
function test(){
        var a=b=123;
}
```

window.a = undifined　　先b赋值但b未定义，参考1，a虽赋值但是a不是全局变量，参考2

window.b = 123

在内部函数中可以调用外部的变量，而外部却不可以用内部的变量

string.fromCharCode　　将ASCII码转化为字符