# 复习

1. 代码执行顺序，先执行骨架HTML（文档树）->css样式->js代码
2. Onclick:鼠标点击前后都在备选元素区域
3. 命名最好用下划线，以数字结尾。单词最好以结构名称，拼音全拼（不要用拼音缩写）

# 笔记

JS三个重点：var function obj

## 一些基本概念

1. 在编程语言中，能够表示并操作的值(value)的类型，称为数据类型(type);
2. 变量(variable)是一个值的符号名称，可以通过变量引用值；
3. 可以把一个值赋值给一个变量(variable)， （变量作用）这样程序在任何需要使用该值的地方，就可以直接引用该变量;

变量（重点）：

1．名词解释： var a1=30

1. 等号右边的:30 代表 有一个数字类型的数据30
2. 等号:赋值
3. var a1: 申明一个空间,给这个空间去一个名字(目的,方便以后使用这个空间或者使用这个空间里面存放的数据)

2．变量取值（打印、调用该变量）

1. 取的是最后一次存放的值
2. 若声明了一个空间,但是没有初始化赋值,想取值就会取到undefined(程序没有报红)





1. 若没有声明一个空间也没有初始化, 想取值，程序报错





总之：取变量(不管是取空间还是空间里的值) ,先看这个空间有没有,在看有没有值

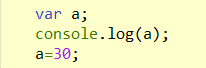
2．存数据（变量赋值）

1. 第一次存,称为初始化,以后的存值就叫赋新值:会覆盖以前的值
2. 存数据到变量时,变量不存在:系统帮你创建一个(全局)变量(运行到这里的时候:按照先后顺序,在script标签的第一行代码,隐式的声明一个变量),在帮你放到这个变量里去





1. 有var: 系统就会在（运行当作用域之前浏览一遍）先在当前作用域的第一行代码隐式的声明一个变量

 相当于



## JavaScript的数据类型分类：

1. 原始类型：数字、字符串、布尔值、null(空)、undefined(未定义)；
2. 对象类型：原始类型之外的类型，如数组、对象、函数等；

## 数字类型（number）

1. JavaScript不区分整数值和浮点数值，所有数字均用浮点数字表示;
2. 直接出现在程序中的数字类型的直接量称为数字直接量；

 30就是直接量

1. 数字直接量分为整型直接量和浮点型直接量；
2. 整型直接量支持十六进制：0x(或0X)作为前缀，其后跟0-9A-Z；
3. 浮点型直接量由整数部分、小数点、小数部分组成,支持指数计数法，如:3.14e13；
4. 数字直接量前面加负号(-)，得到他的负值；
5. 需注意：负号(-)不是数字直接量的一部分，负号(-)是求反运算符；
6. 数字支持的运算符有：+、-、\*、/、%(取余)；
7. Math对象定义的方法可以实现更复杂的操作

Math.random() 生成 [0,1)的一个随机数。

Math.floor()：向下取整（下舍去）

Math.ceil()：向上取整（上进入）

Math.round()：四舍五入

案例1：获取[0,10]的随机整数

Math.round(Math.random()\*10)

案例2：获取[n,m]之间的随机整数

Math.round(Math.random()\*(m-n)+n)

Math.abs() 获取绝对值

Math.max() and Max.min() 获取一组数据中的最大值和最小值

Math.PI 获取圆周率π 的值

Math.pow() : 取幂（两个参数第一个参数为底数，第二个为指数）

案例：Math.pow(10，2) = 100;

Math.sqrt() : 开平方

parseInt(): 将数据类型转换成整型 ~~：取整

案例：将字符串转换为整型，在遇到非数字时就停止不认，截取之前的整数。





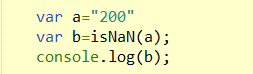
typeof(xxx) :返回xxx数据的类型

正无穷大：Infinity，负无穷大：-Infinity；

1. isNaN()：isNaN=isNot(不是)a(一个)Number(数字),检查一个值是不是数字，

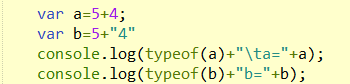
如：isNaN(5)，返回false； isNaN('hello')，返回true；

数值类型的字符串,判定为数字





+: 两个数字(数学的加减运算) 左右只要有一个是字符串(字符串的拼接)





## 浮点数在精度上存在一定的误差

1. 实数有无数个;
2. 但是，JavaScript只能用浮点数表示其中有限的实数，所以，在JavaScript中使用实数的时候，通常只是真实值的一个近似表示；因为JavaScript采用的是二进制表示法，只能用1/2,1/4,1/8…来表示。





1. 如何解决：可以先全部转成整数，运算完后再转回；

## 字符串类型

1. JavaScript通过字符串类型来表示文本;
2. 字符串的索引从0开始，第一个字符的位置是0，第二个是1，以此类推；
3. 长度为1的字符串代表一个字符，JavaScript里面并没有“字符型”；
4. 字符串直接量是指用单引号或双引号括起来的字符序列；
5. " " ' ' 没有区别,都是字符串的直接量的表达方式 :以后遇到了引号里面的引号(一般情况:最外面用单引号,最里面用双引号)
6. 一个字符串可以定义在多行上，建议在非常必要的时候才这么写： 'my name is luzhenchao,\ n am from chengdu,\ n am a it';

## 转义字符

1. 反斜线(\)后面加上一个字符，就不再表示字符的字面含义了，此时称为转义字符；
2. \' 表示对单引号进行转义，这样就可以在单引号括起来的字符串中有单引号了；
3. 常见的转义字符：

\t：水平制表符；

\n：换行符；

\r：回车符；

\"：双引号，而不是字符串分界符；

\'：单引号，而不是字符串分界符；

\\：反斜杠；

## 字符串的一些操作

1. +:左右两边只要有个是string类型,就会把不是字符串类型的转换为(所有的正确的数据都能转化为字符串)字符串,再拼接；
2. 求字符串的长度：str.length；
3. 指定位置的字符：str[2]表示第三个字符;
4. 字符串是固定不变的，对其操作时，会返回一个新的字符串，原始字符串不变；
5. 字符串的更多操作在学习对象及正则表达式时讲；

## 布尔类型

1. 布尔值指对或错、开或关、是或否;
2. 布尔类型只有两个值：true 和 false，是保留字，小写；
3. 布尔值通常用于控制语句中，如if、for、while等；
4. JavaScript的任意值都可以转换为布尔值，可直接使用Boolean()进行转换， 下面的值会转换为false：undefined、null、0、-0、NaN、""





！：表示取反变成布尔值，

！！：表示取布尔值

1. 其他值则转换为true，包括负数、空数组、对象等，好神奇；





## null

1. 是保留字，常用来描述空值;
2. typeof null：返回的是字符串object，也就是说可以把null看成一个特殊的对象；
3. 通常来讲我们把null看成他自有类型的唯一成员；

## undeined

1. undefined表明变量没有初始化；
2. 如果函数没有返回值，则返回undefined；
3. typeof undefined：返回的是字符串undefined；
4. ==认为NULL和undefined是相等的（值相等）；===则返回false；（值相等，类型也相等）