## js第一天笔记

# 就业班笔记

# 一 、Javascript的组成部分

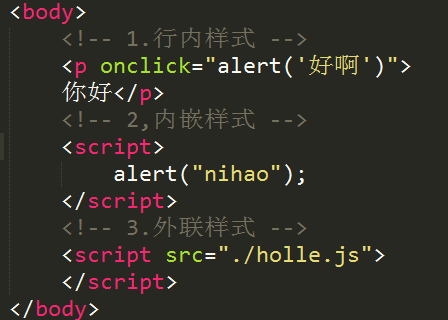
# ECMAscript

DOM：控制的是节点

BOM：控制的是浏览器的工具

## 二、Javascript三种样式：

## 分别是内部样式，内嵌样式，及外联样式



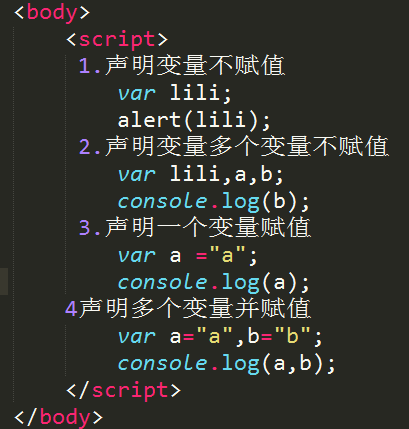
注意：**1. 外联样式的的时候用到script标签的属性 src.**

**2. ./表示的是同级目录下的文件，跟不写是一样的效果，但是为了浏览器读取规定同级目录下也要要写上 ./**

**3.script格式最好是写在body闭合标签的上面，因为浏览器解读是一行行，从上而下解读，这样不会影响HTML的解读**

**4．使用了js外联样式的script标签中不能再使用javascript标签，因为会被忽略。**

## 三、 声明变量的四种方法：



aler（””）;表示的是浏览器中弹出一个窗口

console.log()；在控制台（f12或者右击检查）的console中进行输出

onclick 表示的是当鼠标点击的时候显示的窗口

声明变量格式：

var 变量名=” 变量值”； 声明变量

console.log(变量名)；变量的使用

注意：1.尽量不要声明不给值的变量

2.尽量把所有的变量的声明放在script标签的最上面

3.如果声明多个变量就使用第四点的方式去声明

4.当声明变量但是没有赋值的时候，会出现默认值 undefined未定义

## 四．命名规则

1. 由字母，数字，下划线,$符号组成，不能以数字开头
2. 不能是关键字和保留字（关键字：编程语言中规定不能使用的字符串，保留字：编程语言未来要使用的特殊字符）

## 四．1 命名规范

1. 尽量不要去碰，碰了也没问题只是不好
2. 变量名必须有意义，
3. 没有意义的变量不容易修改，不容易读懂
4. 驼峰命名法：第一个单词小写，后边每个单词的第一个字母大写

小s代表string（字符串）代表

小i代表的是int(整数)

fl 浮点数（小数）

o 代表 object（对象）

b 代表boolean(布尔)

## 五．数据类型：

1、5种简单类型：string（字符串） number（数字） boolean（布尔）undefined(未定义) null（空）

2、1种复杂类型：object（对象）

**String（字符串）**：我们把用双引号或单引号引起来的字符系列，成为字符串

### 字符串的拼接

如果两个字符串相加，那么就会拼接成一个字符串

如果加号两边有一个是字符串，那么也会拼接成一个字符串（不管另一是数字类型，还是其他类型）

如果是减号，那么会发生数字类型转换，这是**隐式转换**

**Number（数字）数据类型**：按照精度分类（是否有小数）分为：整数和小数

数字类型按照进制分类：

**十进制**：

**二进制**

**十六进制**：数字由0-9及a-f一共16个基数构成，必须以0x开头的才是

A 10 b11 c12 d13 e14 f15 如果 ff是255 如果定义的是16进制输出会变成10进制

**八进制**：由0-7这8个基数构成，必须以0开头是8进制 010是8 如果定义的值，大于了进制表示的数字会忽略前面的零按照十进制进行表示

**Boolean（布尔）数据类型**

一般把真假是否定义成布尔值，布尔值只有两个 true（真1，男），false（假 0，女）

Var married=false；

Var num =‘123’；

**Undefined （未定义）数据类型**：当定义了变量没有赋值，就会出现undefiend 函数没有返回值，如果接受了，也会出现undefiend

**Null (空或空对象指针)**：如果定义了一个对象，这个对象想做初始化或对象为初始化就null

如果想让一个变量值为null，只能手写，没有其他方式让他显示为null

null的显示类型为object

Null和null 区别 null是Null数据类型的值

注意 ：大写表示类型，小写表示的是值

转译字符:

字符串的长度：str.length

Console.log(变量名.lenght);会输出几个字符

拼接字符串：在js中我们使用+号拼接字符串

Console.log(a+”he”+b);

除非加号两边都是数字，才是算数运算，否则都是字符串拼接

Typeof运算符，要检测这个数据类型，就使用typeof运算符

格式：console.log（typeof 变量或直接量）

打印函数的类型是 function(函数)

### typeof 变量名或者值 查看是什么数据类型

## 六．数据类型的转换

# 1. 怎么把变量转换成字符串？两种方式

1. toString（）方法：console.log(名字.toString()) undefined和null没有toString转换方法

**转换的都是有意义的数据类型，一般用他比较多**

1. 使用String（）方法 格式：consol.log(String(变量名或者值))

**可以转换没有意义的数据类型，undefined和null**

每种类型的值在控制台中的颜色：

Number类型蓝色

String；类型黑色

Boolean类型是深蓝色

Undefined类型 灰色

Null类型是灰色

**拼接字符串方法：**

如果两个字符串相加，那么就会拼接成一个字符串

如果加号两边有一个是字符串，那么也会拼接成一个字符串（不管另一是数字类型，还是其他类型）

如果是减号，那么会发生数字类型转换，这是**隐式转换**

# 注释

单行注释 [//，只在本行起作用](file:///\\，只在本行起作用)

多行注释 /\*\*/ 一般对象和函数这些的地方使用

# 数字的范围

在js中能表示的数字是有限的，js中有一个最大值，和最小值

**最大值Number.MAX-VALUE**变量 1.79.。。。这是科学记数法

**最小值：Number.MIN-VALU**E常量 5e-324这也是科学记数法 Console.log(Number.MIN-VALUE);

不要去判断浮点数相等，可以判断整数相等

Console.log(1+2==3);返回的是true

##### 2 、把其他类型转换成数字类型

方法Number(名) 格式：console.log(Number(/名))

如果字符串不能转成数字那么就返回NaN

NaN 在js中是一个特殊的数字，

NaN: not a number翻译不是一个数字

判断变量不是一个数字，使用isNaN() 如果不是一个数字返回的是true，如果是一个数字返回的是一个false

Number()方法，如果字符串中带有非数字的字符，不能转化成数字

Number()方法可以把布尔值转变成数字，true变成1 ，flase转变成0

parseInt() 转换成整数，只能转换成整数。方法第一次

parseFloat()转化成浮点数

parseInt()和parseFloat()没有任何区别，只在于parseFloat()可以转换成小数

取正或者取负

不能把布尔变量变成数字

如果是减号，会自动发生转换，把,非数字尝试转换成数字，然后进行算数运算

规定NaN是一个数字，只是比较特殊，他和任何值都不相等包括自己

如果加号两边其中任何一边不时数字那么就是拼接，如果是数字需要相加的话，需要进行数字转换

# 3、把其他类型的变量转换成布尔类型

**方法就一个Boolean(变量名或者值) 1，转化的为true，一般都是true**

**其他类型的值转换为布尔值为false的有 空字符串“”，0， undefined null** NaN

除了转化为false的都会被转化为true

# 操作符（运算符）操作符的操作的操作数可以是数字，变量，或者表达式

##### 算数运算符

##### 算数运算表达式：用算数运算符链接起来的表达式

+ 加

* 减

\*乘

/ 除

% 取余 10%5 取模（取余数）0

##### 逻辑运算符 三种

与**&&** **并且** a和b同时为真才是真，其中一个为假，则为假

或**||** **或者** a和b有一个为真则为真，只有a和b同时为假是才为假

非**！ 取反**

**逻辑运算表达式：由逻辑运算符链接起来的表达式**

**逻辑运算表达式表达的结果也是布尔类型**

### 逻辑运算表达式：用逻辑运算符链接的表达式

##### 复合运算符

复合运算符 **-= += \*= %= /=**

a%=b 相当于 a=a%b 同理以上都是一样

#### 关系运算符

>大于号

<小于号

== 等 只判断值，不判断类型

=== 全等 判断类型也判断值，要判断值和数据类型完全相等

！=不等 和双等类似

！== 不全等 和三个等号类似既要判断值，也要判断类型

>= 大于等于

<=小于等于

在HTML中获取的任何数据都是字符串，包括你字的任何数据，数字，null

数字字面量：你看到的值是多少就是多少，这样的变量的值）

### 关系运算表达式：由关系运算符链接的表达式

### 关系运算表达式的结果是布尔类型

### 一元运算符 ++ --

只能操作操作一个值的操作符

++ 自增运算符

Num++ 后置，意思是先赋值后加加

**后置++或—— 的时候，如果参与了运算，是先参与运算，再自身加一或者减一**

前置++

++num 先自增一，然后赋值

**前置++或——的时候，不管是否参与运算都是先自身加1或者减1,**

## 二元运算符

这个操作符需要操作2个操作数就可以运算，加，减，乘，除都算

## 三元运算符

需要三个操作数，就可以运算，可以是

## 字面量：把一个值直接赋值给一个变量

运算符的优先级

# 流程控制（三种方式）

## 顺序结构：从上到下，从左到右执行就是顺序结构

## 分支结构；if语句，if-else语句，if-else，else if；

### 分支语句if语句

**1**、if语句主要是用用来判断

格式：

**if（表达式） {**

**代码块**

**}**

执行过程： 先判断表达式的结果是true，如果是true则执行大括号中的代码块，如果是false，则不执行

**2**、if-else 语句

语法格式：

**if（表达式）{**

**代码1**

**} else {**

**代码2**

**}**

执行过程：如果表达式结果是true，则执行代码1，否则执行代码2

### 三元表达式

### 三元运算符： ？ ：

三元表达式语法格式：

var 变量=表达式1？表达式2:表达式2

一般if-else分支都可以用三元表达式

执行过程： 如果表达式1是true则执行表达式2，如果是false则执行表达式2

**3**、if-else-if-else-语句

语法格式：

if（表达式1）{

代码1

} else if （表达式2）{

代码2

} else {

代码3

}

执行过程：

如果表达式1是true，则执行代码1，否则判断表达式2

如果表示2是true，则执行代码2，否则执行代码3

**4、分支语句**

**switch –case 语句**

**语法格式：**

**switch （表达式或者是变量）{**

**case:值1：**

**代码1；**

**break；**

**case值2：**

**代码2；**

**break；**

**default:**

**}**

**执行过程：获取表达式的值，和值1比较，如果一样则执行代码1，遇到break，跳出整个语句，后面不执行；如果**

**如果和值1不一样，则和2比较，如果一样，则执行代码2，遇到break，跳出，后面的代码不执行**

**注意：**

## 循环结构：while循环，do while循环，for循环，for in循环；

### 第二 天笔记

### 赋值运算符

赋值运算符就是=

简写方式；

a+=b 表示的是a=a+b 加

a-=b表示的是 a=a-b 减

a\*=b 表示的是a=a\*b 乘

a/=b 表示的是 a=a/b 除

a%=b 表示的是 a=a%b 取模(取余)

表达式和语句：

Var age = 8; 这就是一个表达式

Var num = 3+4；

循环语句：不停不停的执行

判断语句：

程序的三种基本结构：

1. 顺序结构
2. 分支结构{条件语句，判断语句}
3. 循环结构

分支结构：

分支结构语法就是if语句

第一种

如果条件是真，那么会执行花括号的代码

If（条件）{

要做的事情

}

第二种

如果条件满足，那么执行语句1，否则执行语句2

If else

If（条件）{

}else{

}

第三种:

如果

If(条件) {

语句一，满足条件一的时候执行

} else if {

语句2 ，满足条件二的时候执行

} else if{

} else{

}

判断偶数的标志是，这个数字摸于2，余数是否为0，如果是0是偶数，否则为奇数。

三元运算符 (最好不要用，特别复杂的话不直观)

格式： 表达式？语句1：语句2

含义：如果表达式的值为真那么就执行，否则执行语句2

结果：result

简化判断最大值，用三元运算符

大于等于，小于等于，一起作为条件的写法