

21.3.29学习记录

一、隐式类型转换

1、对于非Number类型：

可以使用+任意类型转换成数字类型

二、自增自减

1、自增 ++

通过自增可以使变量在自身的基础上加1，原变量立即增加1

a++: 赋值的话，先取a再加

++a:区别在赋值上，先加再取a

三、逻辑运算符

1、! 非

特别：可以将非布尔值类型转换成布尔值，对一个非布尔值数据两次去反，!! a，即a类型变为布尔值，与Boolean函数一样。

2、&&与

短路操作

3、|| 或

短路操作

以上逻辑运算符对于非布尔值，会先转换为布尔值再运算，并且返回原值；

与运算第一个值为true，返回后面的，第一个值为false返回前面的原值；

或运算则相反!!!

四、赋值运算符

五、关系运算符

任何值和NaN比较都是false

非数值比较会先转换成数值类型;

但是注意, 如果两边都是字符串, 不会转换成数字类型比较, 而是比较的两个字符串的字符编码。而且, 比较字符编码时是一位一位进行比较的。

如果两位一样则比较下一位, 所以借用它来进行英文排序。

在比较两个字符串型的数字式, 一定要转型。

六、在字符串中使用转义字符输入Unicode编码

\u 四位编码;(十六进制)

在网页中输入编码:

&#编码; (这里的编码需要十进制)

七、相等运算符

1、==

字符串1等于数字1, 因为使用==, 如果两个值得类型不同, 会自动转换为相同的类型。大部分情况转换成数字。

undefined与 null是相等的, 因为前者衍生自后者; 但是不全等。

NaN不和任何值相等, 包括他本身。

可以通过isNaN函数判断变量是否是NaN, 是则返回true;

2、!=

类型不相等也会自动转换类型; 但是转换后相等它也会返回false;

- 3、===
全等
- 4、!==
不全等

八、条件运算符

即三元运算符 语法：

条件表达式? 语句1: 语句2;

如果条件表达式返回的非布尔值，则会转换成布尔值;

九、运算符的优先级

- 1、使用 , 可以分割多个语句，一般可以在声明多个变量时使用。

十、

js中可以使用{}来为语句进行分组

同一个{}中的语句我们称为????

一个{}中的语句我们也称为一个代码块

在代码块后面就不用了再编写; 了

js中的代码块，只具有分组的作用，没有其他的用途;

十一、js流程控制语句

- 1、条件判断语句

在执行某个语句之前进行判断，只有条件成立才会执行

if-语句 语法：

if (条件) {语句}

if-else;

if-else if- else;

prompt () 可以弹出一个提示框，该提示框中会带有一个文本框，用户可以在文本框中输入，返回的是字符串；

```
switch(条件表达式){  
    case 表达式:  
        语句;  
        break;  
    default:  
}
```

错题总结{

五种基本数据类型 (Number,String,Boolean,Null,Undefined) ,和一种混合数据类型 (Object)

null的数据类型是object

undefined的数据类型是undefined

1 + null = 1

1 + undefined= NaN

false转为数字为0; true转为数字为1

数字加undefined为NaN;

}