



# JavaScript 基础入门

第 2 天课堂笔记（本课程共 3 天）

前端与移动开发学院

<http://web.itcast.cn>

# 目录

目录 .....	2
一、作业 .....	3
二、布尔值和关系运算符、逻辑运算符 .....	7
2.1 布尔值 .....	7
2.2 关系运算符 .....	8
2.3 逻辑运算符 .....	11
2.4 连比的写法 .....	13
三、if 语句 .....	14
3.1 if 语句初步 .....	14
3.2 多分支的 if 语句和跳楼现象 .....	17
3.3 if 语句的嵌套 .....	20
3.4 if 语句的小知识点 .....	21
四、for 循环语句 .....	22
4.1 认识 for 循环 .....	22
4.2 准确遍历 for 循环 .....	23

# 一、作业

- 1) 【难度★★】用 JS 计算下列算式，并将结果在控制台输出：

$$\frac{123+45 \times 78}{34+543} - 9(324-34)$$

答案：本题考查表达式和运算符

```
1 <script type="text/javascript">
2     var a = (123 + 45 * 78) / (34 + 543) - 9 * (324 - 34);
3     console.log(a);
4 </script>
```

- 2) 【难度★★】世界上表示气温有两种方法：摄氏度和华氏度。两者的关系是：

$$\text{华氏度} = \frac{9}{5} * \text{摄氏度} + 32$$

编写程序，让用户输入摄氏度，向用户提示出对应的华氏温度。

答案：编程，就是有一个整体的步骤：用户输入→计算机计算→输出结果

通过这道题目，我们补充一个小小知识点，叫做“隐式转换”。

加号是有“歧义”的。JS 中+有两层意思：加、连字符

```
1 3 + 5
```

8

```
1 "3" + 5
```

"35"

+这个符号到底是什么意思，必须要看左右的东西是什么类型的。如果都是数字，那么就是加；否则，就是连字符。

但是，其他的运算符，是完全没有歧义的。比如\*、-、/、%。这些运算符，就是用来计算的！

所以，我们的计算机，会帮我们进行一下隐藏的格式转换。

```
1 3 * 5
```

15

```
1 "3" * 5
```

15

```
1 "3" * "5"
```

15

也就是说，如果使用-、\*、/、%，即使不进行格式转换，也是对的。计算机自动帮我们进行“隐式转换”。

### 小测试：

```
1 "4" + 3 - 6
```

37

但是，即使计算机有“隐式转换”，一个靠谱的程序员，一定要自己完成转换。否则，其他人看你的代码，有可能造成误会。

```
1 <script type="text/javascript">
2     //得到用户输入的摄氏温度，用 c 变量来接收
3     //接收的东西是“字符串”，必须通过 parseFloat 才能进行运算
4     var c = parseFloat(prompt("请输入摄氏温度啊"));
5     //计算华氏温度，我们用变量 f 来表示
6     var f = 9 / 5 * c + 32;
7     //显示结果
8     alert("我帮你算出了华氏温度" + f);
9 </script>
```

拓展一下知识：

parseInt() 将字符串转为整数

parseFloat 将字符串转为浮点数（小数）

3) 【难度★★★】用 JS 计算下列算式，并将结果在控制台输出：

$$\left(\frac{23+5^7}{45}\right)^2$$

答案：

```
1 <script type="text/javascript">
2     var a = Math.pow((23 + Math.pow(5,7)) / 45,2);
3     console.log(a);
4 </script>
```

4) 【难度★★★】编写程序，提示用户输入六边形的边长，然后显示它的面积，计算六边形面积的公式是：

$$\text{面积} = \frac{3\sqrt{3}}{2} \text{边长}^2$$

答案：

```
1 <script type="text/javascript">
2     //提示用户输入边长，用变量 a 来接收
3     //记住要 parseFloat 一下。
4     var a = parseFloat(prompt("请输入边长啊"));
5     //计算面积，用变量 s 表示
6     var s = 3 * Math.sqrt(3) / 2 * Math.pow(a,2);
7     //弹出提示框
8     alert("面积是" + s + "，给钱！");
9 </script>
```

5) 【难度★★★★★】如果今天是星期六，那么 1000 天后是星期几？用一条语句计算出来。

答案：

所以我们需要找到共性，我们列表：

几天后	星期几
今天	6
1 天后	7
2 天后	1
3 天后	2
4 天后	3
5 天后	4
6 天后	5
7 天后	6
8 天后	7
9 天后	1
10 天后	2
11 天后	3
.....	
14 天后	6
.....	
21 天后	6
.....	
994 天后	6
995 天后	7
996 天后	1
997 天后	2
998 天后	3
999 天后	4
1000 天后	5
.....	

答案就是星期 5。

994 怎么来的？我们用  $1000/7$  等于 142.8571，表示 142 周零几天就是 1000 天了，我们用  $142*7=994$ 。

上面的做法，实际上太依赖于人脑，就是人的思维和程序杂糅在一起了。老师给出正确答案：

6) 【难度★★★★★】用户输入一个三位数，用程序计算三位数字的和。

比如：

用户输入 155，就弹出 11

用户输入 316，就弹出 10

用户输入 989，就弹出 26

用户输入 678，就弹出 21

注意：用户输入数字的有效性、合法性不用考虑。比如，如果用户输入 34343、“我爱你”，这种情况我们不要去考虑，你就想像，**用户会非常乖**，一定会输入 3 位数的。这是因为我们没有学习 if 语句。

答案：

首先要拿到每一位的位数。也就是说我们需要用我们学到的语句，来完成这个事情，需要一些奇思妙想，需要一些小分析。这个事情，就叫做：

## 算法

算法就是解决问题的方法，经常就是一些“奇思妙想”。

行业内有一句话：

## 算法为王

也就是说，同学们今天开始，就应该改变一下对编程的认识。编程这个东西，和学习英语不一样，不是说你一定要去记忆多少个单词，才多牛逼。而是会解决问题。

**算法：**

比如用户输入 345，怎么分别得到 3？得到 4？得到 5 呢？

怎么得到 3： 345 除以 100，得到 3.45 然后取整，得到 3。简单来说 `parseInt(345/100)`

怎么得到 4： 345 除以 100，余数是 45，除以 10，得到 4.5，取整。简单来说就是 `parseInt(345 % 100 / 10)`

怎么得到 5： 345 除以 10，余数就是 5。简单地说，就是 `345 % 10`

```
1 <script type="text/javascript">
2     //第一步，用户输入数字
3     var num = parseInt(prompt("请输入数字"));
4     //第二步，得到每一位。是我们自己研究的算法
5     var baiwei = parseInt(num / 100);
6     var shiwei = parseInt(num % 100 / 10);
7     var gewei = parseInt(num % 10);
8     //第三步，计算和并显示
9     var sum = baiwei + shiwei + gewei;
10    //显示
11    alert(sum);
12 </script>
```

## 二、布尔值和关系运算符、逻辑运算符

### 2.1 布尔值

我们上节课说了，学习了两种变量的类型数值型、字符串型。

实际上，还有很多变量的类型。我们今天再学习一种，叫做“布尔类型”。

数值型里面的值，有无穷多个，因为世界上所有的数字，都是数值型；

字符串型的值，有无穷多个，因为世界上所有的文字、语句都是字符串型；

布尔类型的值，就两个，true、false。

true 真

false 假

```
1 var a = true;
2 var b = false;
3 console.log(a);    //输出
4 console.log(b);    //输出
5 console.log(typeof a); //输出类型
6 console.log(typeof b); //输出类型
```

true

false

boolean

boolean

布尔值，直接使用就可以了，千万不要加上引号。

```
1 var c = "true";
```

一定要注意拼写，这俩词儿都容易写错：

true 不是 ture

false 不是 flase

## 2.2 关系运算符

上次课，我们学习了一些运算符，都是“数学运算符”。有：

+      -      \*      /      %

实际上，JS 中运算符还有很多，今天我们介绍一种运算符“关系运算符”。

>    大于号  
<    小于号  
>=   大于或等于  
<=   小于或等于  
==   等于  
===   全等于  
!=   不等于  
!==   不全等于

关系运算符，和我们小学学习的意思，一样：

```
1 console.log(5 > 6);  
2 console.log(55 > 6);  
3 console.log(55 > 66);
```

```
false  
true  
false
```

关系运算符，得到的结果都是布尔值，也就是说得到的东西要么是 true，要么是 false

```
1 console.log(-6 < 9);  
2 console.log(-7 < -5);  
3 console.log(5 < 9);  
4 console.log(100 < 5);
```

```
true  
true  
true  
false
```

小时候学习过 $\geq$ ，实际上在 JS 中需要用`>=`来表示，注意中间不能有空格！比如`>=`

```
1 console.log(16 >= 5);        //true  
2 console.log(16 >= 16);       //true  
3 console.log(16 >= 32);       //false
```

```
true  
true  
false
```



判断用户是否及格：

```
1 <script type="text/javascript">
2     //得到用户的成绩
3     var chengji = parseInt(prompt("请输入成绩"));
4     //输出结果:
5     alert(chengji >= 60);
6 </script>
```

小于等于：

```
1 console.log(5 <= 5);    //true
2 console.log(6 <= 10);   //true
3 console.log(6 <= 3);    //false
```

true

true

false

等于

```
1 console.log(5 == 5);    //true
2 console.log(5 == 6);    //false
```

true

false

注意，在 JS 中=符号只有一个意思！表示赋值！！

如果想判断两个东西，是否相等，需要使用符号==

不能用一个等号，来表示等于的关系：

```
1 console.log(3 = 8);
```

Uncaught ReferenceError: Invalid left-hand side in assignment

== 这个符号，还可以验证字符串是否相同：

```
1 console.log("我爱北京天安门" == "我爱北京天安门");
```

==等等不严谨，会将不同类型的东西，转为相同类型进行比较：

```
1 console.log("5" == 5); //true
2 console.log(56 == "56");//true
```

```
true
```

```
true
```

全等于，就是三个等号===

```
1 //全等于
2 console.log("56" === 56); //false
3 console.log(56 === "56"); //false
4 console.log("56" === "56"); //true
5 console.log(56 === 56); //true
```

```
false
```

```
false
```

```
true
```

```
true
```

也就是说，==两个等号，不严谨，"5"和 5 是 true； ===三个等号更为严谨，"5"和 5 是 false。

!= 就是==的反面； !==就是===的反面

```
1 console.log(3 != 8);//true
2 console.log(3 != "3"); //false, 因为 3=="3"是 true, 所以反过来就是 false。
3 console.log(3 !== "3"); //true, 应为 3==="3"是 false, 所以反过来是 true。
```

```
true
```

```
false
```

```
true
```

## 2.3 逻辑运算符

逻辑运算符有三个：

**&&** 与（且）

**||** 或

**!** 非

参与逻辑运算的，都是布尔值。也就是说，只有 `true`、`false` 才能参与逻辑运算，得到的答案，仍然是布尔值。

```
1 // &&表示“且”的意思，都真才真
2 console.log(true && true);      //true
3 console.log(true && false);     //false
4 console.log(false && true);     //false
5 console.log(false && false);    //false
```

`true`

`false`

`false`

`false`

比如：

**“陈伟是男的”是 `true`**

**“1+1 等于 3”是 `false`。**

所以：

**“陈伟是男的”且“1+1 等于 3” 综合结果就是 `false`。**

比如：

**“地球是方的”是 `false`**

**“1+1 等于 3”是 `false`**

所以：

**“地球是方的”且“1+1 等于 3” 综合结果就是 `false`**

或者的意思：

```
1 console.log(true || true);      //true
2 console.log(true || false);     //true
3 console.log(false || true);     //true
4 console.log(false || false);    //false
```

```
true
```

```
true
```

```
true
```

```
false
```

比如：

“陈伟是女的” 是 **false**;

“1+1=2” 是 **true**

所以：

“陈伟是女的” 或者 “1+1=2” 综合结果就是 **true**

比如：

“地球是方的”是 **false**

“1+1=3”是 **false**

所以：

“地球是方的”或者“1+1=3” 综合结果就是 **false**

!表示“非”，没啥好说的，

```
1 console.log(!true);
2 console.log(!false);
```

```
false
```

```
true
```

## 2.4 连比的写法

比如，我们想看看 2 这个数字，是不是在 3 和 15 之间：

```
1 console.log(3<2<15);
```

```
true
```

错误的写法。

这是因为，计算机会先计算  $3 < 2$ ，得到的答案就是 `false`。然后 `false < 15`，`false` 会被隐式转换为 0，所以 `0 < 15` 是 `true`。

应该拆开，中间用一个 `&&` 连接。也就是说：

“3 小于 2” 且 “2 小于 15”。

```
1 console.log(3 < 2 && 2 < 4);
```

```
false
```

判断一个人是否能够考驾照，交通法规定 18~70 岁能够考驾照。

```
1 //得到用户年龄
2 var age = parseInt(prompt("请输入年龄"));
3 //显示结果
4 alert(age >= 18 && age <= 70);
```

## 三、if 语句

### 3.1 if 语句初步

如果……那么……否则……

**if** 就是英语“如果”的意思，**else** 就是“否则”。

举个例子：

```
1  if(明天不下雨){  
2      我就出去玩;  
3  }else{  
4      我就在家写作业;  
5  }
```

公式：

```
1  if(条件表达式){  
2      条件为真的时候做的事情  
3  }else{  
4      条件为假的时候做的事情  
5  }
```

称为 if 语句结构体。

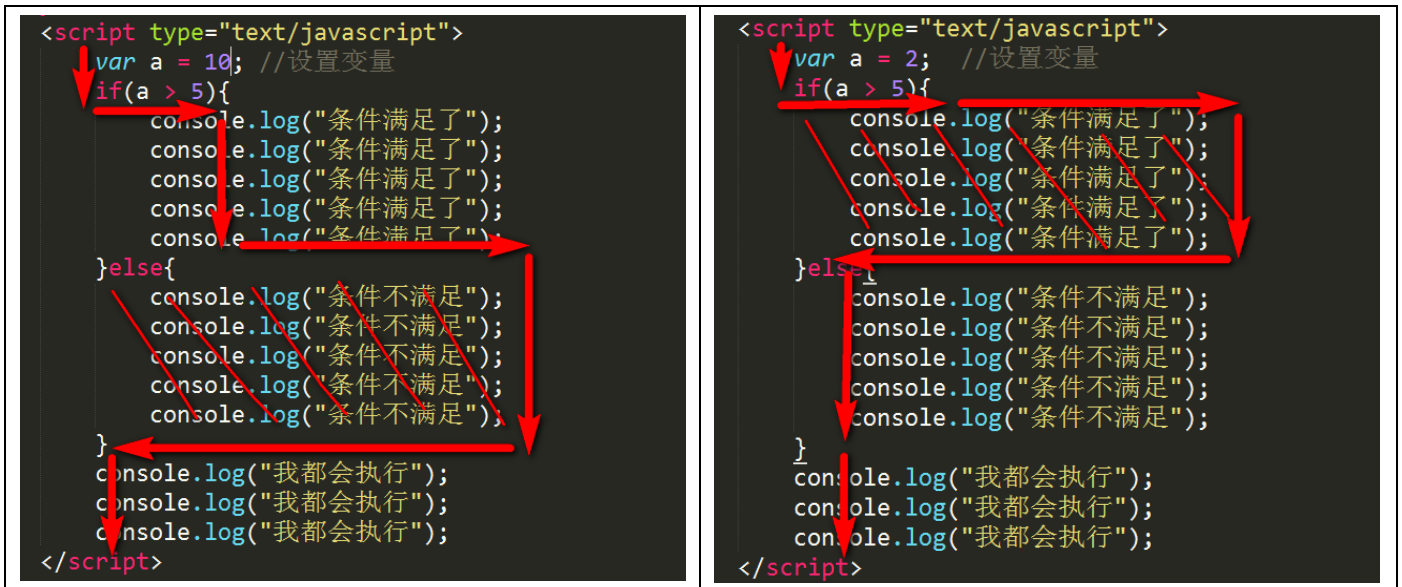
一个程序：

```
1  <script type="text/javascript">  
2      var a = 10;    //设置变量  
3      if(a > 5){  
4          console.log("哈哈");  
5      }else{  
6          console.log("嘻嘻");  
7      }  
8  </script>
```

哈哈

if 语句也成为“选择语句”、“条件判断语句”，我们通过下面的图片来解释。

**殊途同归，走了不一样的路，但是最后都要执行 if 结构体后面的语句。**



条件表达式，要么是 true、要么是 false。绝对在计算机中，不可能出现模棱两可的情况。所以，if 语句让程序有了“选择”，可以在两条路中选择一条。

小题目：

用户输入自己的考试成绩，提示用户是否及格。如果及格了，弹出警告框“恭喜，你及格了”、“不要骄傲啊”。如果没有及格，那么弹出警告框“很遗憾，你没有及格”、“请继续努力啊”。然后都弹出“么么哒”。

答案：

```
1    <script type="text/javascript">
2        //第一步，让用户输入自己的成绩。
3        var score = parseFloat(prompt("请输入你的成绩"));
4        //第二步，判断
5        if(score >= 60){
6            alert("恭喜，你及格了！");
7            alert("不过不要骄傲啊！");
8        }else{
9            alert("很遗憾，你没有及格！");
10           alert("请继续加油啊!! ");
11        }
12        alert("么么哒");
13    </script>
```

小题目：

用户输入自己的年龄，判断用户是否在 18~70 岁，如果在，那么弹出框框“恭喜，可以考驾照”，“加油啊”；否则弹出“年龄不符合要求”。最后，都要弹出“么么哒”。

```
1      <script type="text/javascript">
2          //第一步，得到用户输入的年龄
3          var age = parseInt(prompt("请输入年龄"));
4          //第二步，判断。if 的结构体
5          if(age >= 18 && age <= 70){
6              alert("恭喜，可以考驾照");
7              alert("加油啊");
8          }else{
9              alert("年龄不符合要求");
10         }
11         alert("么么哒");
12     </script>
```

小题目：

判断用户输入的密码是否正确，如果是 123，那么就弹出正确；如果不是，就弹出错误。

```
1      <script type="text/javascript">
2          //第一步，得到用户输入的密码
3          var password = prompt("请输入密码");
4
5          //第二步，判断
6          if(password == 123){
7              alert("正确");
8          }else{
9              alert("错误");
10         }
11     </script>
```



## 3.2 多分支的 if 语句和跳楼现象

用户输入成绩，

如果成绩大于等于 85，那么提示优秀；

否则如果成绩大于等于 70，那么提示良好；

否则如果成绩 60~69，那么提示及格；

否则，不及格

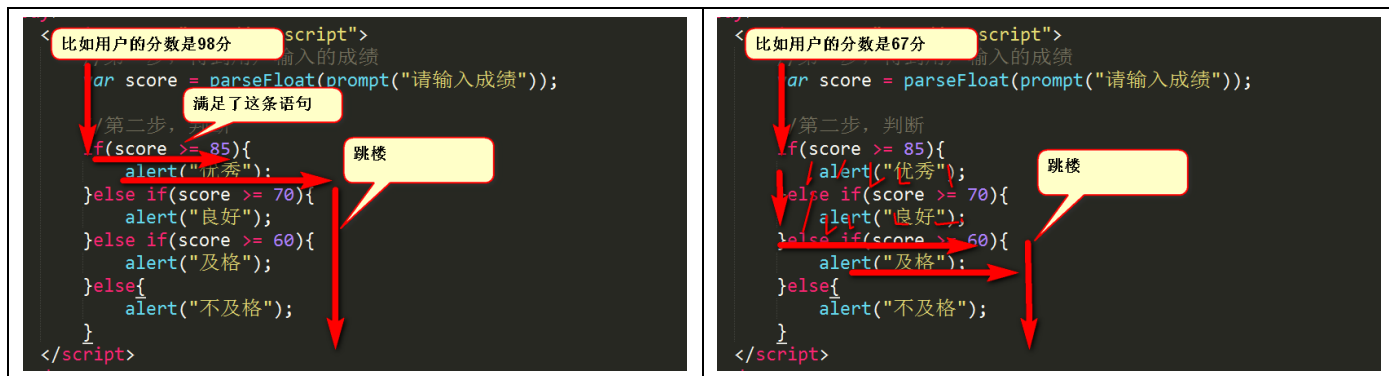
答案：

```
1      <script type="text/javascript">
2          //第一步，得到用户输入的成绩
3          var score = parseFloat(prompt("请输入成绩"));
4
5          //第二步，判断
6          if(score >= 85){
7              alert("优秀");
8          }else if(score >= 70){
9              alert("良好");
10             }else if(score >= 60){
11                 alert("及格");
12             }else{
13                 alert("不及格");
14             }
15     </script>
```

语法：

```
1  if(条件表达式 1){
2      条件 1 为真的时候做的时候
3  }else if(条件表达式 2){
4      条件 1 不满足，条件 2 满足的时候做的事情
5  }else if(条件表达式 3){
6      条件 1、2 不满足，条件 3 满足的时候做的事情
7  }else{
8      全都不满足的时候做的事情
9  }
```

跳楼现象，揭示多分支 if 语句的本质，就是下一个楼层已经暗含之上的楼层都不满足。



小题目：面试题，读下面的程序，试问控制台输出什么？

明天的考试，一定有类似的题目：

```
1      <script type="text/javascript">
2          var a = 10;
3
4          if(a > 5){
5              a = a + 3;
6          }else if(a == 13){
7              a = a + 4;
8          }else if(a == 17){
9              a = a + 5;
10         }else{
11             a = a + 6;
12         }
13
14         console.log(a);
15     </script>
```

小练习：【分组探究作业】根据 BMI（身体质量指数）显示一个人的体型。

BMI 指数，就是体重、身高的一个计算公式。公式是：

$BMI = \text{体重} \div \text{身高的平方}$

比如，老师的体重是 81.6 公斤，身高是 1.71 米。

那么老师的 BMI 就是  $81.6 \div 1.71^2$  等于 27.906022365856163

过轻：低于 18.5

正常：18.5-24.99999999

过重：25-27.99999999

肥胖：28-32

非常肥胖，高于 32

用 JavaScript 开发一个程序，让用户先输入自己的体重，然后输入自己的身高（弹出两次 prompt 框）。计算它的 BMI，根据上表，弹出用户的身体情况。比如“过轻”、“正常”、“过重”、“肥胖”、“非常肥胖”。在程序开发中，充分考虑跳楼现象，不允许出现多余的东西。

方法 1:

```
1    <script type="text/javascript">
2        //第一步，输入身高和体重
3        var height = parseFloat(prompt("请输入身高，单位是米"));
4        var weight = parseFloat(prompt("请输入体重，单位是公斤"));
5
6        //第二步，计算 BMI 指数
7        var BMI = weight / Math.pow(height,2);
8
9        //第三步，if 语句来判断。注意跳楼现象
10       if(BMI < 18.5){
11           alert("过轻，嘻嘻嘻，你不怕被风吹走啊？？多吃点吧！");
12       }else if(BMI < 25){
13           alert("正常，真好!! ");
14       }else if(BMI < 28){
15           alert("过重!还好啊！");
16       }else if(BMI <= 32){
17           alert("肥胖,注意减肥啊！");
18       }else{
19           alert("非常肥胖！");
20       }
21    </script>
```

方法 2:

```
1    <script type="text/javascript">
2        //第一步，输入身高和体重
3        var height = parseFloat(prompt("请输入身高，单位是米"));
4        var weight = parseFloat(prompt("请输入体重，单位是公斤"));
5
6        //第二步，计算 BMI 指数
7        var BMI = weight / Math.pow(height,2);
8
9        //第三步，if 语句来判断。注意跳楼现象
10       if(BMI > 32){
11           alert("非常肥胖");
12       }else if(BMI >= 28){
13           alert("肥胖");
14       }else if(BMI >= 25){
15           alert("过重");
16       }else if(BMI >= 18.5){
17           alert("正常")
18       }else{
19           alert("偏瘦");
20       }
21    </script>
```

### 3.3 if 语句的嵌套

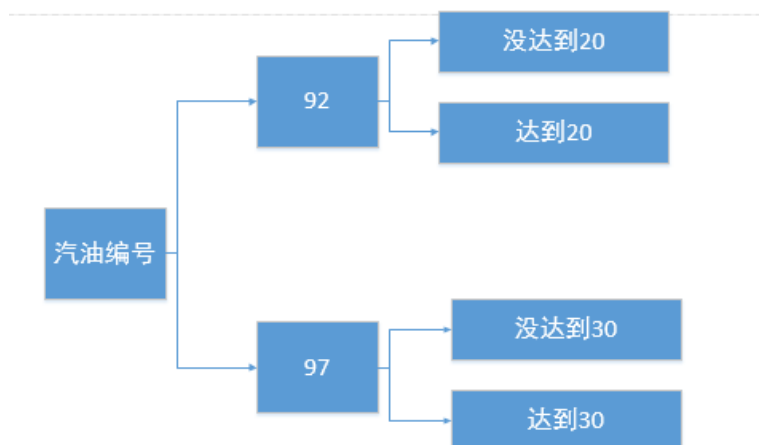
小例子：

一个加油站为了鼓励车主多加油，所以加的多有优惠。

92 号汽油，每升 6 元；如果大于等于 20 升，那么每升 5.9；

97 号汽油，每升 7 元；如果大于等于 30 升，那么每升 6.95

编写 JS 程序，用户输入自己的汽油编号，然后输入自己加多少升，弹出价格。



两层：

```
1 //第一步，输入
2 var bianhao = parseInt(prompt("你想加什么油？填写 92 或者 97"));
3 var sheng = parseFloat(prompt("你想加多少升？"));
4
5 //第二步，判断
6 if(bianhao == 92){
7     //编号是 92 的时候做的事情
8     if(sheng >= 20){
9         var price = sheng * 5.9;
10    }else{
11        var price = sheng * 6;
12    }
13 }else if(bianhao == 97){
14     //编号是 97 的时候做的事情
15     if(sheng >= 30){
16         var price = sheng * 6.95;
17     }else{
18         var price = sheng * 7;
19     }
20 }else{
21     alert("对不起，没有这个编号的汽油！");
22 }
23
24 alert("价格是" + price);
```

小例子：

某个公司要给员工发年终奖，为了奖励老员工，所以工作时间越长，发的越多，规则如下：

工作满 0 年	发月薪的 1 倍月薪年终奖，如果月薪大于 8000，那么就是发 1.2 倍
工作满 1 年	发月薪的 1.5 倍月薪年终奖，如果月薪大于 10000，那么就是发 1.7 倍
工作满 2 年，甚至更多	发月薪的 3 倍月薪年终奖，如果月薪大于 12000，那么就是发 3.2 倍

用 JS 编写程序，让用户输入工作了几年，可以输入 0，然后输入月薪。然后算年终奖。

答案见案例

## 3.4 if 语句的小知识点

**else 部分可以省略。**

```
1 <script type="text/javascript">
2   var a = 10;
3   if(a > 20){
4       alert("这个数字大于 20");
5   }
6 </script>
```

没有 else 部分，就是说没有“否则”，如果条件表达式不满足了，那么就什么都不做。

**如果要做的事情，只有一句话，那么可以省略大括号（初学者尽量不要这么玩儿，一定要把大括号写完整）：**

```
1 <script type="text/javascript">
2   var a = 2;
3   if(a > 5) alert("这个数字大于 5");
4   alert("哈哈哈哈哈");
5 </script>
```

这个程序，a>5 不满足，所以不能弹出“这个数字大于 5”。这是单行 if 语句，省略了大括号，等价于：

```
1 <script type="text/javascript">
2   var a = 2;
3   if(a > 5){
4       alert("这个数字大于 5");
5   }
6   alert("哈哈哈哈哈");
7 </script>
```

所以“哈哈哈哈哈”是能够弹出的。

## 四、for 循环语句

### 4.1 认识 for 循环

在控制台输出 1~100

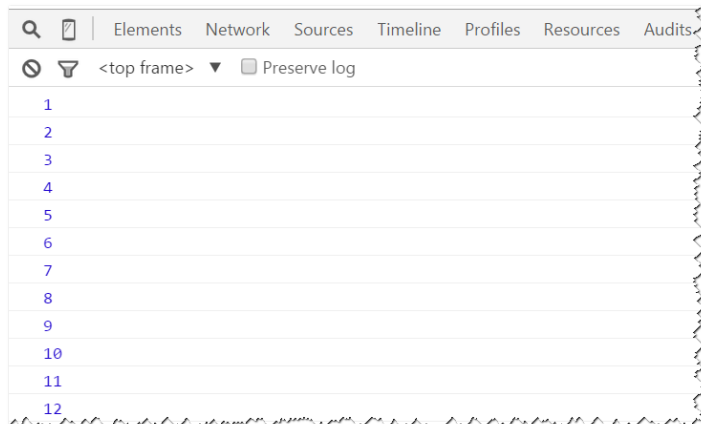
100 行语句:

```
1 console.log(1);
2 console.log(2);
3 console.log(3);
4 console.log(4);
5 console.log(5);
6 console.log(6);
7 console.log(7);
8 console.log(8);
9 .....
10 console.log(100);
```

循环语句，就是执行一些类似的代码，让这些代码的编写变得简化。

程序:

```
1 for(var i = 1; i <= 100; i++){
2     console.log(i);
3 }
```



补充一个知识点，就是

```
1 i++
```

等价于

```
1 i = i + 1;
```

i 自增 1

也就是说

```
1      for(var i = 1 ; i <= 100 ; i++){
2          console.log(i);
3      }
```

等价于:

```
1      for(var i = 1 ; i <= 100 ; i = i + 1){
2          console.log(i);
3      }
```

## 4.2 准确遍历 for 循环

人, 要去分析程序的运行, 走向。

```
1      <script type="text/javascript">
2          for(var i = 1 ; i < 13 ; i = i + 4){
3              console.log(i);
4          }
5      </script>
```

1

5

9

遍历上面的代码:

程序一运行, 将执行 `var i = 1;` 这条语句, 所以 `i` 的值是 1。然后程序会验证一下 `i < 13` 是否满足, `1 < 13` 是真, 所以执行一次循环体 (就是大括号里面的语句)。执行完循环体之后, 会执行 `i=i+4` 这条语句, 所以 `i` 的值, 是 5。

程序会验证一下 `i < 13` 是否满足, `5 < 13` 是真, 所以执行一次循环体 (就是大括号里面的语句)。执行完循环体之后, 会执行 `i=i+4` 这条语句, 所以 `i` 的值, 是 9。

程序会验证一下 `i < 13` 是否满足, `9 < 13` 是真, 所以执行一次循环体 (就是大括号里面的语句)。执行完循环体之后, 会执行 `i=i+4` 这条语句, 所以 `i` 的值, 是 13。

程序会验证一下 `i < 13` 是否满足, `13 < 13` 是假, 所以不执行循环体了, 将退出循环。



```
body>
<script type="text/javascript">
  for(var i = 1 ; i < 13 ; i = i + 4){
    console.log(i);
  }
</script>
body>
html>
```

小练习:

```
1  for(var i = 50 ; i < 40 ; i++){  
2      console.log(i);  
3  }
```

  <top frame

什么也不输出。 因为程序一开始, i=50, 然后会 check 一下 i<40, 为假。不执行

```
1  for(var i = 1 ; i < 10 ; i = i + 3){  
2      i = i + 1;  
3      console.log(i);  
4  }
```

2

6

10

程序一开始 i=1,所以进入循环体, i 变为 2, 输出 2。

然后执行 i=i+3, 所以 i 等于 5 了, 满足 i<10, 所以进入循环体了, i 变为 6, 输出 6

然后执行 i=i+3, 所以 i 等于 9 了, 满足 i<10, 所以进入循环体了, i 变为 10, 输出 10

然后执行 i=i+3, 所以 i 等于 13 了, 不满足 i<10, 所以不执行循环体。

```
1  for(var i = 1 ; i <= 10 ; i++){  
2  
3  }  
4  console.log(i);
```

11

程序一开始, i 等于 1, 满足 i<=10, 进入循环体, 但是循环体没有语句, 不会输出。

i++执行。 所以, 程序一直当 i 为 10 的时候, 又执行了一次 i++, 所以 i 等于 11。不满足 i<=10 了, 退出循环, 输出 11.

```
1  for(var i = 1 ; i < 7 ; i = i + 3){  
2  
3  }  
4  console.log(i);
```

7



```
1      for(var i = 1 ; i < 10 ; i = i + 1){
2          if(i % 2 == 0){
3              i = i * 2;
4          }
5          console.log(i);
6      }
```

1  
4  
5  
12  
>

```
1  for(var i = 1 ; i > 0 ; i++){
2      console.log(i);
3  }
```

死循环。 循环永远停不下来

for 循环灵活:

```
1  var i = 3;
2  for(; i < 10 ; i++){
3      console.log(i);
4  }
```

3  
4  
5  
6  
7  
8  
9

```
1      for(var i = 3 ; i < 20 ; i = i + 2){
2          if(i % 3 == 2){
3              i = i + 1;
4          }else{
5              i = i + 2;
6          }
7          console.log(i);
8      }
```

5  
9  
12  
15  
18