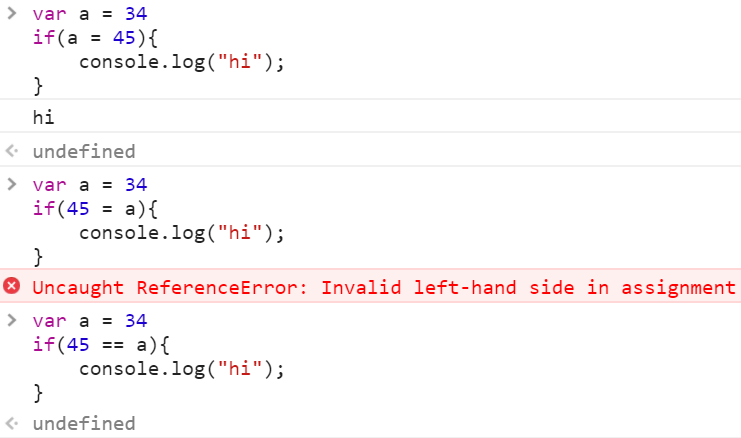
**第五章**

**赋值运算符**

* 注意=与==（表达式要发反写，有什么好处）



**算数运算符**

* 算数运算符与类型转换  
  "1"+"2"; //"12"  
  "1"+2; //"12"  
  1+{}; //"1[object Object]"  
  true+true; //2   
  "5"-2; //3

一元运算符（++、--）  
var x = "1";  
console.log(++x); //2 注意++和--的隐式类型转换  
var x = "1";  
console.log(x+=1);//11

**关系运算符**

* == 与 ===  
  == （如果类型不同，先转换再比较，注：引用类型到基本类型的转换方向）  
  === （若类型不同则false，若类型相同则判断同 ==）直接进行类型比较  
  回顾值类型与引用类型的比较结果（3===3、{}==={}、NaN===NaN）
* != 与 !==  
  ！=（相当于==的逆运算）  
  ！==（先判断类型，若类型不同则返回true，相当于===的逆运算）