# 传统求素数方法

*双重for循环遍历，一个个判断*

for(int i=2;i<=n;i++){  
 boolean flage=true;  
 for(int j=2;j<i;j++ ){  
 if(i%j==0){  
 flage=false;  
 break;  
 }  
 }  
 if(flage){  
 System.out.println("素数"+i);  
 }  
}

# 埃式筛法

素数：大于1的自然数中，除了1和它本身以外不再有其他[因数](https://baike.baidu.com/item/%E5%9B%A0%E6%95%B0/9539111" \t "https://baike.baidu.com/item/%E8%B4%A8%E6%95%B0/_blank)的自然数

原理：

求n以内的素数，将根号n以内素数的倍数剔除掉剩下的就是素数。

2，3，4，5，6，7，8，9，10，11，12，13，14，15，16，17，18，19，20

实现步骤：

1. 定义一个长度为n+1的bool数组，将下标为[2,n]的元素记为true,true表示素数
2. 判断1到根号是否为素数，如果为该下标下为素数则将其倍数的数对应的boolean数组下标变为false

理解难点：将2的倍数，4，6，8....变为素数后，第一层循环判断该数是否为素数时结果为false,跳过

实现代码：

public static void sieve2(int n){  
 //定义布尔类型数组用于判断该索引下的数是否为数组  
 //默认把所有当做素数  
 boolean[] isPrime=new boolean[n+1];  
 int cnt=0;  
 for(int i=2;i<=n;i++){  
 isPrime[i]=true;  
 }  
 for(int i=2;i<Math.sqrt(n);i++){  
 if(isPrime[i])//当前下标数为质数，则它的倍数不为质数  
 {  
 for(int j=2\*i;j<=n;j+=i){  
 if(isPrime[j]==true){  
 isPrime[j]=false;  
 }  
 }  
 }  
  
 }  
 for(int i=2;i<=n;i++){  
 if(isPrime[i])  
 {  
 System.out.println("素数"+i);  
 cnt++;  
 }  
 }  
 System.out.println("共有"+cnt+"个质数");  
}