**[C++ 随机数生成](http://www.cppblog.com/Niino/archive/2008/08/16/59042.html)**

标准库<cstdlib>（被包含于<iostream>中）提供两个帮助生成伪随机数的函数：

函数一：int rand(void)；  
从srand (seed)中指定的seed开始，返回一个[seed, RAND\_MAX（0x7fff）)间的随机整数。

函数二：void srand(unsigned seed)；  
参数seed是rand()的种子，用来初始化rand()的起始值。

可以认为rand()在每次被调用的时候，它会查看：  
1） 如果用户在此之前调用过srand(seed)，给seed指定了一个值，那么它会自动调用  
srand(seed)一次来初始化它的起始值。  
2） 如果用户在此之前没有调用过srand(seed)，它会自动调用srand(1)一次。

根据上面的第一点我们可以得出：  
1） 如果希望rand（）在每次程序运行时产生的值都不一样，必须给srand(seed)中的seed一个变值，这个变值必须在每次程序运行时都不一样（比如到目前为止流逝的时间）。  
2） 否则，如果给seed指定的是一个定值，那么每次程序运行时rand（）产生的值都会一样，虽然这个值会是[seed, RAND\_MAX（0x7fff）)之间的一个随机取得的值。  
3） 如果在调用rand()之前没有调用过srand(seed)，效果将和调用了srand(1)再调用rand()一样（1也是一个定值）。

举几个例子，假设我们要取得0～6之间的随机整数（不含6本身）：

例一，不指定seed：  
for(int i=0;i<10;i++){   
ran\_num=rand() % 6;  
cout<<ran\_num<<" ";  
}  
每次运行都将输出：5 5 4 4 5 4 0 0 4 2

例二，指定seed为定值1：  
srand(1);  
for(int i=0;i<10;i++){   
ran\_num=rand() % 6;  
cout<<ran\_num<<" ";  
}  
每次运行都将输出：5 5 4 4 5 4 0 0 4 2  
跟例子一的结果完全一样。

例三，指定seed为定值6：  
srand(6);  
for(int i=0;i<10;i++){   
ran\_num=rand() % 6;  
cout<<ran\_num<<" ";  
}  
每次运行都将输出：4 1 5 1 4 3 4 4 2 2  
随机值也是在[0,6）之间，随得的值跟srand(1)不同，但是每次运行的结果都相同。

例四，指定seed为当前系统流逝了的时间（单位为秒）：time\_t time(0)：  
#include <ctime>  
//…  
srand((unsigned)time(0));  
for(int i=0;i<10;i++){   
ran\_num=rand() % 6;  
cout<<ran\_num<<" ";  
}  
第一次运行时输出：0 1 5 4 5 0 2 3 4 2  
第二次：3 2 3 0 3 5 5 2 2 3  
总之，每次运行结果将不一样，因为每次启动程序的时刻都不同（间隔须大于1秒？见下）。

关于time\_t time(0)：

time\_t被定义为长整型，它返回从1970年1月1日零时零分零秒到目前为止所经过的时间，单位为秒。比如假设输出：  
cout<<time(0);  
值约为1169174701，约等于37（年）乘365（天）乘24（小时）乘3600（秒）（月日没算）。

另外，关于ran\_num = rand() % 6，

将rand()的返回值与6求模是必须的，这样才能确保目的随机数落在[0,6)之间，否则rand()的返回值本身可能是很巨大的。  
一个通用的公式是：  
要取得[a,b)之间的随机整数，使用（rand() % (b-a)）+ a （结果值将含a不含b）。  
在a为0的情况下，简写为rand() % b。

最后，关于伪随机浮点数：

用rand() / double(RAND\_MAX)可以取得0～1之间的浮点数（注意，不同于整型时候的公式，是除以，不是求模），举例：  
double ran\_numf=0.0;  
srand((unsigned)time(0));  
for(int i=0;i<10;i++){   
ran\_numf = rand() / (double)(RAND\_MAX);  
cout<<ran\_numf<<" ";  
}  
运行结果为：0.716636，0.457725，…等10个0～1之间的浮点数，每次结果都不同。

如果想取更大范围的随机浮点数，比如1～10，可以将  
rand() /(double)(RAND\_MAX) 改为 rand() /(double)(RAND\_MAX/10)  
运行结果为：7.19362，6.45775，…等10个1～10之间的浮点数，每次结果都不同。  
至于100，1000的情况，如此类推。