<https://blog.csdn.net/lina_acm/article/details/51915028?readlog>

**K&RC**  
  
1978年，丹尼斯·里奇（Dennis Ritchie）和BrianKernighan合作出版了《C程序设计语言》的第一版。书中介绍的C语言标准也被C语言程序设计师称作“K&RC”，第二版的书中也包含了一些ANSI C的标准。K&RC主要介绍了以下特色：   
结构（struct）类型   
长整数（long int）类型   
无符号整数（unsigned int）类型   
把运算符=+和=-改为+=和-=。因为=+和=-会使得编译器不知道用户要处理i = +10还是i =-10，使得处理上产生混淆。   
  
即使在后来ANSI C标准被提出的许多年后，K&RC仍然是许多编译器的最低标准要求，许多老旧的编译仍然运行K&RC的标准。   
  
[编辑]   
**ANSI C 和 ISO C**  
主条目：**ANSI C**  
  
1989年，C语言被 ANSI 标准化（ANSIX3.159-1989）。标准化的一个目的是扩展K&RC。这个标准包括了一些新特性。在K&R出版后，一些新特性被非官方地加到C语言中。   
void 函数   
函数返回 struct 或 union 类型   
void \* 数据类型   
  
在ANSI标准化自己的过程中，一些新的特性被加了进去。ANSI也规定一套了标准函数库。ANSI ISO（国际标准化组织）成立ISO/IEC JTC1/SC22/WG14 工作组，来规定国际标准的C语言。通过对ANSI标准的少量修改，最终通过了 ISO9899:1990。随后，ISO标准被 ANSI 采纳。   
  
传统C语言到ANSI/ISO标准C语言的改进包括：   
增加了真正的标准库   
新的预处理命令与特性   
函数原型允许在函数申明中指定参数类型   
一些新的关键字，包括 const、volatile 与 signed   
宽字符、宽字符串与字节多字符   
对约定规则、声明和类型检查的许多小改动与澄清   
  
WG14工作小组之后又于1995年，对1985年颁布的标准做了两处技术修订（缺陷修复）和一个补充（扩展）。下面是 1995年做出的所有修改：   
3 个新的标准库头文件 iso646.h、wctype.h 和 wchar.h   
几个新的记号与预定义宏，用于对国际化提供更好的支持   
printf/sprintf 函数一系列新的格式代码   
大量的函数和一些类型与常量，用于多字节字符和宽字节字符   
  
[编辑]   
**C99**  
主条目：**C99**  
  
在ANSI的标准确立后，C语言的规范在一段时间内没有大的变动，然而C++在自己的标准化建立过程中继续发展壮大。《标准修正案一》在1995年为C语言建立了一个新标准，但是只修正了一些C89标准中的细节和增加更多更广得国际字符集支持。不过，这个标准引出了1999年ISO9899:1999的发表。它通常被成为C99。C99被ANSI于2000年3月采用。   
  
**在C99中包括的特性有：**   
对编译器限制增加了，比如源程序每行要求至少支持到 4095 字节，变量名函数名的要求支持到 63 字节（extern 要求支持到31）   
预处理增强了。例如：   
宏支持取可变参数 #define Macro(...) \_\_VA\_ARGS\_\_   
使用宏的时候，参数如果不写，宏里用 #,## 这样的东西会扩展成空串。（以前会出错的）   
支持 // 行注释（这个特性实际上在C89的很多编译器上已经被支持了）   
增加了新关键字 restrict, inline, \_Complex, \_Imaginary,\_Bool   
支持 long long, long double \_Complex, float \_Complex这样的类型   
支持 <: :> <%%> %: %:%: ，等等奇怪的符号替代，D&E里提过这个   
支持了不定长的数组。数组的长度就可以用变量了。声明类型的时候呢，就用 int a[\*]这样的写法。不过考虑到效率和实现，这玩意并不是一个新类型。所以就不能用在全局里，或者 struct union里面，如果你用了这样的东西，goto 语句就受限制了。   
变量声明不必放在语句块的开头，for 语句提倡这么写 for(int i=0;i <100;++i)就是说，int i 的声明放在里面，i 只在 for 里面有效。   
当一个类似结构的东西需要临时构造的时候，可以用（type\_name）{xx,xx,xx} 这有点像 C++的构造函数   
初始化结构的时候现在可以这样写：   
struct {int a[3]，b;} hehe[] =  { [0].a = {1},[1].a = 2 };   
struct {int a, b, c, d;} hehe =  { .a = 1, .c = 3,4, .b = 5}  // 3,4 是对 .c,.d赋值的   
字符串里面，\u 支持 unicode 的字符   
支持 16 进制的浮点数的描述   
所以 printf scanf 的格式化串多支持了 ll / LL（VC6 里用的 I64）对应新的 long long类型。   
浮点数的内部数据描述支持了新标准，这个可以用 #pragma 编译器指定   
除了已经有的 \_\_line\_\_ \_\_file\_\_ 以外，又支持了一个 \_\_func\_\_可以得到当前的函数名   
对于非常数的表达式，也允许编译器做化简   
修改了对于／% 处理负数上的定义，比如老的标准里 -22 / 7 = -3, -22 % 7 = -1 而现在 -22 / 7 =-4, -22 % 7 = 6   
取消了不写函数返回类型默认就是 int 的规定   
允许 struct 定义的最后一个数组写做 [] 不指定其长度描述   
const const int i；将被当作 const int i；处理   
增加和修改了一些标准头文件，比如定义 bool 的 <stdbool.h>定义一些标准长度的 int 的 <inttypes.h> 定义复数的<complex.h> 定义宽字符的<wctype.h> 有点泛型味道的数学函数<tgmath.h> 跟浮点数有关的<fenv.h>。<stdarg.h> 里多了一个 va\_copy 可以复制 ...的参数。 <time.h> 里多了个 struct tmx 对struct tm 做了扩展   
输入输出对宽字符还有长整数等做了相应的支持   
  
但是各个公司对C99的支持所表现出来的兴趣不同。当GCC和其它一些商业编译器支持C99的大部分特性的时候，微软和Borland却似乎对此不感兴趣。