作者：代兴　邮箱：503268771@qq.com　Vx 公众号：东方架构师

函数名的规则

函数名表示函数的名称，源文件中称为函数名，汇编文件中称为函数的符号名。
函数名的规则：
通用规则。由字母、数字、下划线、美元符组成，首字母不能是数字。不同的编程语言可能有细微差别。
编译器把函数名编译为符号名。CPU 和汇编，使用符号名，不使用函数名。
符号名和函数名，可能相同，也可能不相同。
C 语言，函数名和符号名相同，不支持函数重载。
C++语言，函数名和符号名一般不相同，支持函数重载。符号名的生成规则复杂，依赖命名空间、类名、方法名、参数类型等。
动态库，函数调用，使用符号名查找对应的函数。

用 C 和 C++对比函数名

编写 C 程序： name_c.c

```
// 一个 C 函数。有入参和返回值。
int study_at_2023(double param1, int param2, char *param3)
{
    return 1;
}
```

编写 C++程序： name_cpp.cpp

```
using namespace std;

// 一个 C++函数。有入参和返回值。
int study_at_2023(double param1, int param2, char *param3)
{
    return 1;
}
```

把代码编译为汇编文件：

```
# 编译 C 程序。
gcc -S name_c.c -o name_c.s
# 编译 C++程序。
g++ -S name_cpp.cpp -o name_cpp.s
```

比较符号名：
C 程序的函数名为 study_at_2023，汇编文件 name_c.s 对应的符号名为 study_at_2023。说明，函数名和符号名相同。

```
    .globl  study_at_2023
    .type   study_at_2023, @function
study_at_2023:
```

```
.LFB0:
    .cfi_startproc
    pushq   %rbp
    .cfi_def_cfa_offset 16
```

C++程序的函数名为 study_at_2023，汇编文件 name_cpp.s 对应的符号名为_Z13study_at_2023diPc。说明，函数名和符号名不相同。

```
    .globl  _Z13study_at_2023diPc
    .type   _Z13study_at_2023diPc, @function
_Z13study_at_2023diPc:
.LFB0:
    .cfi_startproc
    pushq   %rbp
    .cfi_def_cfa_offset 16
```

问题：2 个函数的名称相同，为什么符号名不同？
符号名的生成，与编程语言有关。符号名是函数的唯一标识。C 语言、C++语言，生成唯一标识的方法不相同。
C 语言规定函数名唯一表示一个函数。如果 2 个函数的函数名相同，但是入参不同，则认为是同一个函数，即 C 语言不支持函数重载。
C++语言规定命名空间、类名、方法名、入参唯一表示一个函数。C++语言支持函数重载，所以符号名更长更复杂。

比较函数名相同，入参不同

编写 C 程序： many_name_c.c

```
// 函数，同名
int name_c_1(int age)
{
    return 1;
}

// 函数，同名
int name_c_1(double age)
{
    return 1;
}
```

编译 C 程序：

```
gcc -S many_name_c.c -o many_name_c.s
```

编译错误。因为 2 个函数的符号名都为 name_c_1，符号重名了。

```
many_name_c.c:10:5: error: conflicting types for ‘name_c_1’
 int name_c_1(double age)
     ^
many_name_c.c:4:5: note: previous definition of ‘name_c_1’ was here
 int name_c_1(int age)
     ^
```

编写 C++程序：many_name_cpp.cpp

```
using namespace std;

// 函数，同名
int name_c_1(int age)
{
    return 1;
}

// 函数，同名
int name_c_1(double age)
{
    return 1;
}
```

编译 C++程序：

```
g++ -S many_name_cpp.cpp -o many_name_cpp.s
```

编译成功。
查看 C++的汇编文件：many_name_cpp.s 看到 2 个函数对应的 2 个符号。说明 C++支持函数重载。
符号名_Z8name_c_1i 对应 函数 `int name_c_1(int age)` 。
符号名_Z8name_c_1d 对应 函数 `int name_c_1(double age)` 。
符号名末尾，i 表示 int，d 表示 double。

```
    .globl  _Z8name_c_1i
    .type   _Z8name_c_1i, @function
_Z8name_c_1i:
.LFB0:
    .cfi_startproc

    .globl  _Z8name_c_1d
    .type   _Z8name_c_1d, @function
_Z8name_c_1d:
.LFB1:
    .cfi_startproc
```