

打开VS2019 → 创建新项目 → 空项目

视图 → 解决方案管理器 .c 源文件 .h 头文件

选中源文件 右击添加新建项 .c 命名 ✓ .cpp 为 C++ X
按 C++ 编译

#include <stdio.h>

若库打招呼

int main ()

C 语言一定要有 main 函数

{

printf ("hehe\n");

printf 为库函数 专门用来打印数据

return 0; ~~数据~~

VS 2019, Ctrl+F5 运行代码

}

调试一开始执行 (不调)

正常返回 0 不正常非 0 (习惯)

fn + Ctrl + F5

0 为执行过程 main 函数为程序入口 只有一个

三注释选中行 三取消注释

Ctrl + K + C

Ctrl + K + U

数据类型

小数 - 浮点型

float 单精度
double 双精度

6.66 * 10 66.6

整数 - 整型

int

Q 一字符 char

短整型 short

长整型 long

更长整型 long long

printf ("%d\n", 100); %d 为整型

printf ("%d\n", sizeof(char)); char 类型数据在内存中所占大小

可能报错 %d → %zu

sizeof 返回的数值
是整型

unsigned short

int long

long long

float

double

1
2
4
4
8
4
8

二进制 $\begin{matrix} \text{bit} & \text{bit} \\ \boxed{1} & \boxed{0} \end{matrix}$ 8个 = BYTE 1kb = 1024 byte

mb gb tb pb

$\text{sizeof}(\text{long}) > \text{sizeof}(\text{int})$

类型越多空间使用更灵活

要存20向内存申请个空间

`int age = 20;` age为变量 类型向内存申请空间创建变量

`double price = 66.6;` 精度高用 double 低用 float

`int b = 20;` 全局变量

`int main()`

{

`short age = 20;` 1/年龄 定义变量顺便给变量赋值

`int high = 180;` 1/身高

`float weight = 88.5;`

`int a = 10;` 局部变量

`int a = 100;`

`int main()`

{

`int a = 10;`

`printf("a=%d\n", a);` 局部优先

`return 0`

但不要局部全局名一样

写一个代码，计算2个整数的和

```
int main ( )
```

```
{
```

```
int num1 = 0;
```

```
int num2 = 0; //初始化 不初始化 num 为随机值
```

```
scanf ("%d %d", &num1, &num2); //取地址) 输入
```

```
int int sum = num1 + num2;
```

```
printf ("%d\n", sum)
```

报错 To disable deprecation use _____

第1行 #define _____

scanf_s 只是 VS 编译器函数 其它编译器不认识

VS 安装路径有 new c++ file.cpp 建 .c 或 .cpp 再
都拷贝这个文件

下个 everything 搜 new c++ file.cpp 右击打开路径
打开记事本改 复制到桌面保存再拷回去

这个名字可以在哪使用 哪就是作用域

```
int main ( int a = 10 全局变量 )
```

```
{
```

```
{ ← 生命周期开始 进入作用域
```

```
int a = 10;
```

```
printf ("a = %d\n", a);
```

```
} → --- 结束
```

```
printf ("a = %d\n", a);
```

```
return 0;
```

```
}
```

^{a 的局部作用域}

若另一个源文件 `int a=10;` 这个文件要用 `a`
要声明来自外部符号

`extern int a;`

全局变量作用域是整个工程，生命周期是整个程序生命周期