

## 文件格式

对于电脑来说 所有的文件最后都会被 解码 (decode) 成 二进制, 不论数字, 字符, 图片, 声音。但是同一串二进制的数字

120 => 1111000(2)

1. 可以表示 数字 120
2. 可以表示字符 x
3. 可以表示颜色 深蓝色 <https://g.co/kgs/xqmxU1A>

RGB R (0 - 255) G (0 - 255) B (0 - 255)

算一下有多少种颜色

文字: ASCII GBK Unicode UTF-8 (如何显示)

ASCII 重点 含有普通的英文字母 数字 标点符号 和 特殊符号 空格 ESC 回车 等等

unicode = utf-8 含有世界上所有的文字和符号 - 中文汉字 韩文 俄文 甲骨文? 表情包

## 编程语言概念

c++ 需要先把代码编译成一个二进制文件 binary ^ 然后再运行这个文件,

变量(variable): 在程序用来存储数据的可以变化的量

命名: 由 数字 字母 和下划线组成 不能以数字开头 也不能是关键字

e.g. x, x\_1, x1, varName\_x1, \_name\_x

下列名字不能作为变量名:

1x, 23y, return(关键字), int(关键字)

函数, 库, 赋值, 表达式vs语句

运算符号

+ - \* / 加减乘除

% 取余数

+= -= \*= /=

x = x+3; 可以写成 x+=3;

== != < <= > >= && || 表示比较关系

++ -- 递增 递减

```
// x = 3
```

```
x++; // x = 4
```

## 输入和输出

```
cin >> x; // 要求用户输入一个值给变量x
```

```
cout << x; // 打印变量x的值到屏幕上
```

```
#include <iostream> // 引入iostream库 含有 cout cin endl 等特殊关键词
```

```
/*
```

库是一组函数的集合，我们可以通过引入库来使用库里面的函数

可以认为是一个文件，里面包含了很多函数的定义，公式可以让我们使用

但是在别的文件中我们需要引入这个文件才能使用这个文件中的函数

```
*/
```

```
#include <string> // 引入string库 含有 string 类型
```

```
using namespace std; // namespace是一个命名空间 std是一个命名空间的名字
```

```
/*
```

命名空间是为了解决命名冲突问题的，我们可以在不同的命名空间中定义相同的函数名

例如 在两个不同的文件里 我都定义了 hello这个函数 但是他们的内容是不一样的

这时候 我需要告诉编译器 我要使用哪个命名空间里的 hello函数 他才能分清楚

```
*/
```

```
/*
```

函数声明

函数需要

1. 函数返回类型 int

2. 函数名 main

3. 参数列表 在括号里定义()

4. 函数体 在花括号里定义{}

5. 返回值 return 0; 也可以不返回 用void 作为函数返回类型

函数的作用可以是得到一个计算的返回值 也可以是执行一些操作

例如 打印，读取文件，删除文件，创建连接，发送请求等等

```
*/
```

```
int main(){ // 主函数
```

```
    short x = 3; // 定义变量 声明变量类型 变量名 = 值;
```

```
    int y = 3;
```

```
    long z = 3;
```

```
    cout << sizeof(x) << endl; // 打印 x 占用的（字节）2 endl 是换行符 我们也可以使用 \n 字符代替
```

```
    cout << sizeof(y) << endl; // 打印 y 占用的（字节）4
```

```
    cout << sizeof(z) << endl; // 打印 z 占用的（字节）8
```

`char c = 'a';` // c++ 的char类型只能 存储 ascii表格里的字符 ascii拓展表格一共是256个字符 正好占一个字节

```
cout << sizeof(c) << endl; // 打印 c 占用的 (字节) 1
```

`bool a = true;` // bool 只有 true(对) 或者 false(错)两种值

```
cout << a << endl; // 代表 true, 0 代表 false c++ 通常会打印 a 的值 1
```

```
cout << sizeof(a) << endl; // 打印 a 占用的 (字节) 1
```

`/* ----作业-----`

作业 分析一下为什么 bool 类型占用的字节是1 明明只有两个值 似乎只需要一个比特bit就够了

`*/`

`string s = "hello @李雷雷😄😭🚗";` // string 类型是一个特殊的c++ 类型 不同于 char他可以包含 unicode字符

```
cout << s.size() << endl; // 打印 s 占用的 (字节) 28
```

`/* ----作业-----`

分析一下string s 当中哪些字符各占用了多少字节

`*/`

```
return 0; // 返回0 代表程序正常退出
```

```
};
```