

- 第零章 - 算法思维
- 第一章 - 代码框架
- 第二章 - 变量
 - 变量的类型
 - 变量命名规则
 - 变量的定义
- 第 章 - 指针

第零章 - 算法思维

在互联网时代背景下，计算机技术逐渐被越来越多人所使用，而编程就是让计算机听懂人命令最直接和最有效的方式。那何为编程？编程可以简单的理解为程序员为解决特定问题，按照自己的思路，遵循计算机语言规则下编写程序的过程。一般也将进行编写代码的编码员称为“码农”。计算机编程学的是编程的概念，但注重的是计算机的思维，比如循环、函数、算法。它的理论性强，通常强调要以最有效的方式去解决问题，而且是在多场景通用的解决方法。学习编程就要学会对所见信息进行组织、分析，即培养逻辑思维，找到事物的相互关联。利用程序解决问题时，通常要完成以下三个方面：

- 组织数据：就是需要考虑如何组织被处理的数据、中间结果和最终结果。数据的组织方式也称为数据结构。例如：利用一维数组存放学生的身高；
- 设计算法：在确定数据结构的基础上，设计处理数据的方法和步骤；
- 编写程序：根据数据结构和算法编写程序代码，反复调试和测试直到程序符合要求。编程注重的是算法思维，而不是计算机的语言规则，因为计算机语言规则是死的。当然编程也是严谨的将这些问题思考清楚，我们在掌握计算机语言规则的基础上就能编写出多种正确的程序代码。最后仍需强调一下，我们实际需要解决的问题不断在变化，学习编程需要读者注重过程的思考，而非结果本身。

第一章 - 代码框架

- 可以用 `include` 引用头文件。如用 `#include<bits/stdc++.h>` 可以引用 `bits` 文件夹下的 `stdc++.h` 文件。
- 可以用 `using namespace` 使用命名空间。如用 `using namespace std;` 可以使用名叫 `std` 的命名空间。注意后面有分号 `;`。注:除了预处理(开头有 `#` 的)外，理论上来说，所有代码末尾都有分号 `;`。
- `main` 函数是程序的运行的开端。程序是最先运行 `main` 函数的。

- 在函数里写 `return 0;` 可以给函数返回值0。(main函数里的 `return 0;` 可以不写，编译器会自动添加。)
- `//` 一般用于单行注释，注释以 `//` 开始，直到行末为止。
- `/* ... */` 一般用于多行注释，注释以 `/*` 开始，以 `*/` 终止。

```
#include <bits/stdc++.h> //引用头文件
using namespace std; // 使用命名空间
int main() { // 良好的码风可以让程序看起来赏心悦目。
    return 0; // 养成好习惯。退出程序。
}
```

第二章 - 变量

变量顾名思义是会变得量。

变量的类型

数据类型	数据范围
int	-2147483648 ~ 2147482647
long long	$-2^{63} \sim 2^{63} - 1$
char	-128 ~ 127
bool	0或1
double	15 位有效数字

变量命名规则

1. 变量名只能出现字母、数字、下划线。
2. 第一个字符不能是数字，例如 `1a` 是不合法的变量名。
3. 不能使用 C++ 已经定义好的有特殊含义的单词。例如 `int`
4. 区分大小写，例如 `a1` 和 `A1` 是两个不同的变量。

变量的定义

第 章 - 指针

- 指针就是内存地址，指针变量是用来存放内存地址的变量。
- 在 C++ 中，可以通过指针来改变变量的值。指针是一个变量，它存储了另一个变量的地址。通过指针，我们可以访问到这个变量，从而修改它的值。
- 下面是一个简单的示例，演示如何用指针来改变变量的值

```
#include <iostream>
using namespace std;
int main()
{
    string s = "I love c++!";
    string *p = &s;
    cout << *p << '\n';
    *p = "We love c++!";
    cout << *p << '\n';
    return 0;
}
```

- 这段代码定义了一个字符串s，它的值为 I love c++!。然后定义了一个指针变量p，它存储了s的地址。通过使用指针p，代码输出了s的值，即 I love c++!。接着，代码通过指针p修改了s的值为 We love c++!，并再次输出了s的值，即 We love c++!。