## 实例一：程序基本结构 (最简单的 "Hello World")

**Python 代码**

```python
# 直接写，所见即所得
print("Hello, C++!")
```

**C++ 代码**

```cpp
// C++ 程序在基本的程序框架下写代码，这个框架基本是固定的结构，记住即可。
#include <iostream>
using namespace std;

int main() {
    cout << "Hello, Python!" << endl;
    return 0;
}
```

### 区别点解析：

- **框架**：Python 几乎没有框架，直接写功能。C++ 必须把所有代码放进 `int main() { ... }` 这个主函数“容器”里。
- **引用**：C++ 需要用 `#include <iostream>` 来引入“输入输出”的功能模块。
- **语法**：C++ 语句结尾必须有分号 `;`。`endl` 用于换行。

---

## 实例二：变量与数据类型 (声明一个变量并使用)

**Python 代码**

```python
# Python 是动态类型
# 变量的类型由赋给它的值决定

my_variable = 100
print("它现在是一个数字:", my_variable)

my_variable = "一百"
print("它现在变成了一个字符串:", my_variable)
```

**C++ 代码**

```cpp
#include <iostream>
#include <string> // 使用 string 类型，需要引入 string 工具箱中的字符串模块

using namespace std;

int main() {
    // C++ 是静态类型
    // 必须提前声明变量的类型，语法为：数据类型 变量名 = 初始化值；其中 = 初始化值 的部分可以省略

    int my_number = 100;
```

```
    cout << "它是一个整数: " << my_number << endl;

    // 如果是C++，下面这行代码会直接编译失败!
    // my_number = "一百";

    string my_text = "一百";
    cout << "这是一个字符串: " << my_text << endl;

    return 0;
}
```

### 区别点解析：

- Python 的变量 `my_variable` 可以随意改变类型。C++ 的变量 `my_number` 在被声明为 `int` 后，就永远只能存储整数。
- **声明**：C++ 必须用 `类型 名字;` 的格式先"声明"变量的存在，才能使用。如 `int my_number;` 和 `string my_text;`。

---

## 实例三：输入与输出

**Python 代码**

```
name = input("请输入你的名字: ")
age_str = input("请输入你的年龄: ")

# input() 得到的是字符串，需要手动转换
age = int(age_str)

print("你好,", name, "! 你明年就", age + 1, "岁了。")
```

**C++ 代码**

```
#include <iostream>
#include <string>

using namespace std;

int main() {
    string name; // 准备一个装字符串的盒子
    int age;     // 准备一个整数的盒子

    cout << "请输入你的名字: ";
    cin >> name;

    cout << "请输入你的年龄: ";
    // cin 会自动将用户的输入转换为整数!
    cin >> age;

    cout << "你好, " << name << "! 你明年就 " << age + 1 << " 岁了。" << endl;

    return 0;
}
```

### 区别点解析：

- **输入类型**：Python 的 `input()` 永远返回字符串，需要手动转换。C++ 的 `cin` 会根据你提供的变量类型（`int age`）自动尝试转换用户的输入，非常方便。
- **输出串联**：Python 的 `print()` 用逗号 `,` 分隔多项。C++ 的 `cout` 使用 `<<` 符号来像链条一样串联多项。

---

## 实例四：条件判断

### 例：输入年龄，根据是否成年 以及 是否有许可 来判断能否进入一个俱乐部

注：**是否有许可**是通过一个布尔类型变量 `has_permission` 来手动设置的，变量值设置为 `true` 代表有许可，否则是无许可。

**Python 代码**

```python
age = 20
has_permission = False

# 使用 and, or, not 逻辑词
# 使用冒号(:)和缩进定义代码块
if age >= 18 and not has_permission:
    print("你已成年，但没有许可。")
elif age < 18:
    print("你还未成年。")
else:
    print("你可以进入。")
```

**C++ 代码**

```cpp
#include <iostream>

using namespace std;

int main() {
    int age = 20;
    bool has_permission = false; // 布尔值是小写

    // 使用 &&, ||, ! 逻辑符号
    // 使用圆括号()包裹条件，大括号{}定义代码块
    if (age >= 18 && !has_permission) {
        cout << "你已成年，但没有许可。" << endl;
    }
    else if (age < 18) { // 注意是 else if
        cout << "你还未成年。" << endl;
    }
    else {
        cout << "你可以进入。" << endl;
    }

    return 0;
}
```

**区别点解析：**

- **逻辑词 和 逻辑符号 不同**：Python 的 `and`, `or`, `not` 在 C++ 中对应的是 `&&`, `||`, `!`。
- **代码块**：Python 靠**冒号+缩进**。C++ 靠**大括号** `{...}`。
- **条件包裹**：C++ 的 `if` 和 `else if` 后面的判断条件**必须**用圆括号 `()` 包起来。
- **关键字不同**：Python 的 `elif` 在 C++ 中是 `else if`。Python 的 `True/False` 在 C++ 中是 `true/false`。