

ASCII（美国标准信息交换代码）是一种字符编码标准，用于表示文字、数字和其他符号。ASCII 码使用 7 位或 8 位（二进制）表示字符，通常在计算机中以 8 位（1 字节）存储。

1. ASCII 码基础

- 范围：
 - 标准 ASCII：0 至 127（7 位），包括控制字符（0-31）和可打印字符（32-126）。
 - 扩展 ASCII：128 至 255（8 位），包括其他字符和符号（例如，西欧语言中的特殊字符）。
- 控制字符（0-31）：这些字符主要用于控制输出设备（例如，换行符、回车符、响铃符等）。
- 可打印字符（32-126）：包括数字、字母、标点符号等。

2. ASCII 表

十进制	二进制	十六进制	字符	描述
0	0000000	0x00	NUL	空字符
32	0100000	0x20	空格	空格
48	0110000	0x30	0	数字0
65	1000001	0x41	A	大写字母A
97	1100001	0x61	a	小写字母a
122	1111010	0x7A	z	小写字母z
127	1111111	0x7F	DEL	删除字符

3. 在 C++ 中使用 ASCII

3.1 打印 ASCII 字符

你可以通过字符的 ASCII 值打印字符：

```
#include <iostream>
using namespace std;

int main() {
    // 打印 ASCII 码为 65 的字符 ('A')
    char ch = 65;
    cout << "ASCII 65对应的字符是： " << ch << endl;

    return 0;
}
```

输出：

ASCII 65对应的字符是： A

3.2 通过字符获取 ASCII 值

你也可以通过字符获取其 ASCII 值：

```
#include <iostream>
using namespace std;

int main() {
    char ch = 'A';
    int ascii_value = int(ch); // 将字符转换为其 ASCII 值
    cout << "'A' 的 ASCII 值是: " << ascii_value << endl;

    return 0;
}
```

输出：

```
'A' 的 ASCII 值是: 65
```

3.3 使用 ASCII 控制字符

你可以在 C++ 程序中使用 ASCII 控制字符来实现特定功能。例如，换行符（`\n`）和回车符（`\r`）：

```
#include <iostream>
using namespace std;

int main() {
    // 使用换行符（ASCII 10）
    cout << "这是第一行\n这是第二行" << endl;

    // 使用回车符（ASCII 13）
    cout << "Hello\rworld" << endl;

    return 0;
}
```

输出：

```
这是第一行
这是第二行
world
```

3.4 字符范围检查

你可以检查字符是否属于某个特定的 ASCII 范围。例如，检查字符是否为字母或数字：

```
#include <iostream>
using namespace std;

int main() {
    char ch = 'A';

    // 检查是否是大写字母
```

```

    if (ch >= 'A' && ch <= 'Z') {
        cout << ch << " 是大写字母" << endl;
    }

    // 检查是否是数字
    if (ch >= '0' && ch <= '9') {
        cout << ch << " 是数字" << endl;
    }

    return 0;
}

```

输出：

```
A 是大写字母
```

3.5 将字母转换为大写或小写

ASCII 码提供了一种简单的方式来将字母转换为大写或小写。例如，字母 `a` 的 ASCII 值是 97，而字母 `A` 的 ASCII 值是 65，它们之间相差 32。所以可以通过加减 32 来进行转换。

```

#include <iostream>
using namespace std;

int main() {
    char lower = 'a';
    char upper = 'A';

    // 将小写字母转换为大写
    lower = lower - 32;
    // 将大写字母转换为小写
    upper = upper + 32;

    cout << "小写字母转换为大写：" << lower << endl;
    cout << "大写字母转换为小写：" << upper << endl;

    return 0;
}

```

输出：

```
小写字母转换为大写：A
大写字母转换为小写：a
```

4. 扩展 ASCII 和 Unicode

- **扩展 ASCII (128-255)**：虽然标准 ASCII 只使用 7 位，但许多计算机系统支持扩展 ASCII，使用 8 位来表示字符。这些扩展字符用于支持其他语言的特殊字符、符号等。你可以通过查看字符对应的扩展 ASCII 值来获得这些字符。
- **Unicode**：对于支持更多语言字符的需求，通常使用 Unicode 编码（例如 UTF-8、UTF-16）。在 C++ 中，可以使用 `wchar_t` 和 `std::wstring` 来处理 Unicode 字符。

5. ASCII 与 C++ 字符串操作

你可以利用 ASCII 码对字符串进行处理。例如，遍历字符串并修改字符：

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    string str = "Hello world!";

    // 将字符串中的所有字母转换为大写
    for (int i = 0; i < str.length(); i++) {
        if (str[i] >= 'a' && str[i] <= 'z') {
            str[i] = str[i] - 32; // 转换为大写
        }
    }

    cout << "转换后的字符串: " << str << endl;

    return 0;
}
```

输出：

```
转换后的字符串: HELLO WORLD!
```

6. 总结

- **ASCII 码**是计算机内部表示字符的一种标准方式，范围从 0 到 127，扩展 ASCII 码则支持 128 到 255 的字符。
- 在 C++ 中，字符实际上是其 ASCII 值的整数表示，`char` 类型本质上是一个整数类型，存储 ASCII 值。
- 使用 ASCII 码，你可以轻松实现字符的转换、范围检查和字符串操作等。

希望这些内容能帮助你更好地理解和使用 ASCII 码在 C++ 中的应用。如果有更多问题，欢迎继续提问！