



字符串类

字符串类 `string` 是不同于字符数组的字符串处理方式。使用字符串类，需要包含以下头文件：

```
1. #include<string>
```

在学习过程中，必须要分清楚 `string` 类和字符数组处理字符串的异同。



字符串类：输入输出（cin/cout）

可以使用 cin 和 cout 来读写 string 类型。

使用 cin 读入 string 类型，会忽略开头的制表符、换行符、空格，当再次碰到这类字符就停止（并不会读取这类字符）。

【参考程序】

```
1. #include<string>
2. #include<iostream>
3. using namespace std;
4. int main()
5. {
6.     string s;
7.     int total = 0;
8.     while (cin>>s)
9.     {
10.         total++;
11.         cout << total << ' ' << s << endl;
12.     }
13.     return 0;
14. }
```

【输入样例】

```
Hello, xiaozhi!
Hello
```

【输出样例】

```
1 Hello,
2 xiaozhi!
3 Hello
```



字符串类：输入输出（getline）

读取包含空格类字符的一行：

```
1. getline(cin, s);
```

其中 `s` 是字符串类的对象。

【参考程序】

```
1. #include<string>
2. #include<iostream>
3. using namespace std;
4. int main()
5. {
6.     string s;
7.     int total = 0;
8.     while (getline(cin, s))
9.     {
10.         total++;
11.         cout << total << ' ' << s << endl;
12.     }
13.     return 0;
14. }
```

【输入样例】

```
Hello, xiaozhi!
Hello
```

【输出样例】

```
1 Hello, xiaozhi!
2 Hello
```



字符串类：构造 string 对象和初始化

【参考程序】

```
1. #include<iostream>
2. #include<string>
3. using namespace std;
4. int main()
5. {
6.     string s; //空串
7.     string s1 = "Hello, Xiaozhi!"; //初始化为"Hello, Xiaozhi!"
8.     string s2("OK"); //初始化为"OK"
9.     string s3(s1); //用 s1 初始化 s3
10.    string s4(4, 'c'); //初始化为 4 个字符'c'
11.    cout << s << endl << s1 << endl << s2 << endl << s3 << endl << s4 << endl;
12.    return 0;
13. }
```



字符串类：string 类字符元素的访问

1. `s[i]` //返回 `s` 中索引 `i` 处字符的引用，不查是否出界
2. `s.at(i)` //返回 `s` 中索引 `i` 处字符的引用，查是否出界

【参考程序】

```
1. #include<iostream>
2. #include<string>
3. using namespace std;
4. int main()
5. {
6.     string s = "Hello, Xiaozhi!";
7.     cout << s[1] << ' ' << s.at(4) << endl;
8.     cout << s[20] << endl;
9.     cout << s.at(20) << endl;
10.    return 0;
11. }
```



字符串类：重载运算符

【参考程序】

```
1. #include<iostream>
2. #include<string>
3. using namespace std;
4. int main()
5. {
6.     string s1;
7.     string s2;
8.     s1 = "Hi";
9.     s2 = s1; //s2 成为 s1 的拷贝
10.    cout << s2 << endl;
11.    s1 += s2; //s2 的字符数据连接到 s1 的尾部
12.    cout << s1 << endl;
13.    cout << s1 + s2 << endl; //将 s2 连接到 s1 的尾部, s1 不改变
14.    //基于字典序的比较, 返回布尔值, true 或 false
15.    cout << (s1 == s2) << endl;
16.    cout << (s1 != s2) << endl;
17.    cout << (s1 < s2) << endl;
18.    cout << (s1 > s2) << endl;
19.    cout << (s1 <= s2) << endl;
20.    cout << (s1 >= s2) << endl;
21.    return 0;
22. }
```



字符串类：常用成员函数

length()	求长度
size()	求长度
empty()	判断是否为空串
substr()	求子串
erase()	删除若干个字符
insert()	插入字符
replace()	替换字符
find 系列函数	查找子串

表1. 字符串类常用函数



字符串类：求长度函数 length / size

【参考程序】

```
1. #include<iostream>
2. #include<string>
3. using namespace std;
4. int main()
5. {
6.     string s = "Hello, Xiaozhi!";
7.     cout << s.length() << endl;
8.     cout << s.size() << endl;
9.     return 0;
10. }
```




字符串类：判断空串函数 empty

【参考程序】

```
1. #include<iostream>
2. #include<string>
3. using namespace std;
4. int main()
5. {
6.     string s1 = "Hello, Xiaozhi!";
7.     string s2 = "";
8.     cout << s1.empty() << endl;
9.     cout << s2.empty() << endl;
10.    return 0;
11. }
```