

String类型标准库函数

1. 构造函数与赋值

- 构造函数：支持从字面量、其他字符串或子串构造。

```
string s1;           // 空字符串
string s2("Hello");  // "Hello"
string s3(s2);        // 复制 s2
string s4(5, 'a');    // "aaaaa"
```

- 赋值 = 和 assign() :

```
s1 = "World";        // s1 变为 "World"
s2.assign(s1, 0, 3);  // s2 变为 "Wor"
```

2. 容量操作

- 长度与容量：

```
s.size();           // 返回字符数（等同于 s.length()）
s.empty();           // 判断是否为空
s.capacity();        // 返回当前分配的存储空间
```

- 调整大小 resize() :

```
s.resize(10, 'x');   // 调整长度为 10，不足部分填充 'x'
```

- 清空 clear() :

```
s.clear();           // 清空内容，s 变为空字符串
```

3. 元素访问

- 下标访问 `[]` 和 `at()` :

```
char c1 = s[1];    // 不检查越界（高效）
char c2 = s.at(1); // 越界时抛出异常
```

- 首尾字符 `front()` 和 `back()` :

```
char first = s.front(); // 首字符
char last  = s.back();  // 尾字符
```

4. 修改操作

- 追加 `append()` 和 `+=` :

```
s.append(" World"); // 追加字符串
s += "!";           // 追加字符或字符串
```

- 插入 `insert()` :

```
s.insert(5, " C++"); // 在位置 5 插入 " C++"
```

- 删除 `erase()` :

```
s.erase(5, 3); // 从位置 5 删除 3 个字符
```

- 替换 `replace()` :

```
s.replace(0, 5, "Hi"); // 替换前 5 个字符为 "Hi"
```

- 交换 `swap()` :

```
s1.swap(s2); // 交换 s1 和 s2 的内容
```

5. 字符串操作

- 提取子串 `substr()` :

```
string sub = s.substr(6, 5); // 从位置 6 提取 5 个字符
```

- C 风格字符串 `c_str()` 和 `data()` :

```
const char* ptr = s.c_str(); // 返回以空字符结尾的字符数组
```

6. 查找与比较

- 查找 `find()` 系列:

```
size_t pos = s.find("lo"); // 查找子串, 返回位置或 string::npos  
pos = s.find_first_of("aeiou"); // 查找第一个元音字母
```

- 比较 `compare()` 或运算符:

```
if (s1 == s2) { ... } // 直接比较  
int res = s1.compare(s2); // 返回 0 表示相等
```

7. 其他实用功能

- 大小写转换 (需结合 `<algorithm>`) :

```
transform(s.begin(), s.end(), s.begin(), ::tolower); // 转小写
```

- 类型转换 (非成员函数) :

```
int num = stoi("123"); // 字符串转整数  
string str = to_string(3.14); // 数值转字符串
```

