\*[std::String-CSDN](https://blog.csdn.net/zy2317878/article/details/79056289)

## 构造函数

1. 默认构造函数，构造一个空的string
2. 复制一个string来构造
3. 复制string的一部分来构造，起始位置加偏移量，或者用两个迭代器
4. 一个长度加一个元素，类似于vector的初始化

std::string s0 ("Initial string"); //根据已有字符串构造新的string实例  
std::string s1; //构造一个默认为空的string  
std::string s2 (s0); //通过复制一个string构造一个新的string  
std::string s3 (s0, 8, 3); //通过复制一个string的一部分来构造一个新的string。8为起始位置，3为偏移量。  
std::string s4 ("A character sequence"); //与s0构造方式相同。  
std::string s5 ("Another character sequence", 12); //已知字符串，通过截取指定长度来创建一个string  
std::string s6a (10, 'x'); //指定string长度，与一个元素，则默认重复该元素创建string  
std::string s6b (10, 42); // 42 is the ASCII code for '\*' //通过ASCII码来代替s6a中的指定元素。  
std::string s7 (s0.begin(), s0.begin()+7); //通过迭代器来指定复制s0的一部分，来创建s7

## operator

1. 重载 = ： 用于拷贝构造string
2. 重载 + ： 用于拼接两个string

std::string str1, str2, str3;  
str1 = "Test string: "; // c-string //通过=运算符来给已创建的string“赋值”  
str2 = 'x'; // single character  
str3 = str1 + str2; // string   
//注意这里重载了"+",string类的"+"可以理解为胶水，将两个string类型连接起来了

## 元素获取&容量

|  |  |
| --- | --- |
| 函数 | 功能 |
| size() | 返回string的长度 |
| length() | 和size一样，但其实是c风格的，stl加入的size |
| capacity() | 重新分配内存之前，string所能包含的最大字符数 |
| max\_size() | 一个string最多能够包含的字符数 |
| resize() | 将string的长度更改为一个指定参数的长度，除去最后超出的或者在最后插入空 |
| reserve() | 更改了string的容量capaticy，并没有实际更改size |
| operator[] | 返回字符串中位置pos处字符的引用 |
| at() | 和operator的区别在于会检测下标是否越界，然后抛出异常 |
| back();front() | 返回对字符串第一个/最后一个字符的引用；begin()和end()返回迭代器 |

## 元素更改

|  |  |
| --- | --- |
| 函数 | 功能 |
| operator+=;append() | 通过在当前值的末尾添加其他字符来扩展字符串 |
| push\_back() | 与append()区别在于一次只能push一个字符 |
| assign() | 为字符串分配一个新值，括号里可以是构造函数的任一形式 |
| insert() | 在由pos指示的字符前插入附加字符，第二个参数可以是构造函数的形式 |
| erase() | 擦除字符串中的字符，输入两个数字则第一个为位置第二个为消除数量；输入两个迭代器则消除区间；输入一个迭代器则只消除一个字符； |
| replace() | 用字符串替换原string的部分内容，前两个参数是区间(数字或迭代器)；剩余参数是string的构造 |
| swap() | 交换两个字符串的内容 |

## **string-string operations**

|  |  |
| --- | --- |
| 函数 | 功能 |
| c\_str() | string转char\*，返回一个临时指针，可以作为函数参数等，但是不能对该指针操作 |
| copy() | 从string类型对象中至多复制n个字符到字符指针p指向的空间中，第二个参数为n，第三个为起始位置，默认为0 |
| find() | 在字符串中查找内容，找到则返回查找内容第一个字符的位置，找不到返回string :: npos |
| substr() | 返回一个新构造的字符串对象，子字符串是从字符位置pos开始的对象的一部分；可以1个参数即子串从pos到end，2个参数则子串从posa延续len，如果延续len超过end自动取end |
| compare() | 返回比较结果；1个参数可以只输入一个字符串；3个参数可以输入自己的开始位置和长度加字符串；5个参数可以输入自己的位置长度加字符串和字符串的开始位置加长度 |
| std::atoi() | const char\*转int，a\_toi(s.c\_str())，超过边界则输出边界 |
| std::stoi() | const string&转int，超过int边界会报错 |
| std::to\_string() | 数字转string，整数浮点数都可以 |

## istringstream字符串流对象用法

//1.分割字符串
  
string data("abcd,efgh");
  
vector<string> strs; //分割完的字符串vector
  
istringstream sin(data); //将字符串转为string流对象
  
string str; //用于存放每次分割下来的字符串
  
while(getline(sin, str, ',')) {
  
 strs.push\_back(str);
  
}
  
//2.string转int或double
  
istringstream is("3.14");
  
double k;
  
is>>k;