

## مقدمه بر رشته ها ( string ) در سی پلاس پلاس

نویسنده : رضا آشتیانی

رشته ها در سی پلاس پلاس مثل آرایه ای کاراکتر ها هستند تو زبان سی با این تفاوت که تو هر زمان می تونه طولشون افزایش یا کاهش پیدا کنه .

برای استفاده از string ها کافیه کتابخانه مورد نظرش ( string ) رو include کنیم .

### #include < string >

برای اینکه بتونیم از اسامی استاندارد موجود در این کتابخانه استفاده کنیم باید "اسم فضای استاندارد" رو هم به سیستم معرفی کنیم .

### using namespace std;

حالا می تونیم به string بسازیم و ارزش استفاده کنیم .

```
string a;
```

با این دستور به string داریم به اسم a و می تونیم از توابع موجود در این کتابخانه برای a استفاده کنیم .  
به طور خلاصه اعضای این کلاس را بررسی می کنیم .

### ثابت عددی npos

ثابتی عددی است برابر با بزرگترین اندیسی که رشته می تواند داشته باشد .

### operator =

با این تابع می تونید مستقیما به یه رشته مقدار بدید .

```
string a = "abcd";
```

الان a شامل رشته abcd شده ، آگه دوباره مقدار دهی کنید ، مقدار قبلی رشته از دست میره و مقدار جدید جایگزین میشه .

```
a = "dcba";
```

الان رشته شامل dcba شده و مقدار قبلی رشته کاملا از دست رفته و امکان بازگشت وجود نداره .

### operator []

همون طور که گفتیم string ها آرایه هایی هستند با طول متغیر ، پس می تونیم مثل آرایه ها باهاشون رفتار کنیم .

```
a[2] = 'x';
```

الان رشته a میشه dcxa .

### operator +

این عملگر دو رشته رو با هم جمع می کنه ، منظور از جمع اینه که رشته دوم به آخر رشته اول چسبیده میشه .

```
string s, t;
```

```
s = "123";
```

```
t = "456";
```

```
cout << s + t << endl;
```

این دستور رشته 123456 رو چاپ می کنه یا این مثال :

```
s = s + t;
```

```
cout << s << endl;
```

این دستور ها معادل دستور بالا عمل می کنه .

این توابع همون طور که از اسمشون معلومه operator هستند و نحوه فراخوانی خاص خودشون رو دارند ، در ادامه با بقیه توابع آشنا میشیم که برای یه string صدا زده میشن ؛ برای استفاده باید از '! استفاده کنیم .

### size()

این تابع اندازه ( طول ) آرایه رو بر می گردونه ، این تابع ورودی نداره و یه عدد بر می گردونه .

```
cout << a.size() << endl;
```

این دستور عدد 4 رو چاپ می کنه .

## clear()

این تابع رشته رو خالی می کنه ، همچنین حافظه تخصیص یافته به string رو آزاد می کنه .

```
a.clear();  
cout << a.size() << endl;
```

این دستور عدد 0 رو چاپ می کنه .

## begin()

این دستور به اشاره گر به نقطه شروع string بر می گردونه .

```
a = "abcdef";  
cout << *a.begin() << endl;
```

دستور دوم کاراکتر 'a' رو چاپ می کنه .

چون پیاده سازی این اشاره گر ها خطی هستش ، می تونید به عدد رو با این اشاره گر جمع کنید تا به خونه های دیگه رشته دسترسی داشته باشید .

```
cout << *(a.begin() + 2) << endl;
```

این دستور کاراکتر 'c' رو چاپ می کنه .

## end()

این تابع به اشاره گر به محل پایان رشته بر می گردونه ، لازم به ذکر است محل پایان رشته ، اولین خونه بعد از خونه آخر رشته خواهد بود .

```
cout << *(a.end() - 1) << endl;
```

این دستور کاراکتر 'f' رو چاپ می کنه .

## resize()

این تابع طول رشته رو به اندازه دلخواه می تغییر میده . دو نوع پیاده سازی برای این تابع وجود داره  
1 - پیاده سازی اول فقط طول رو تغییر میده .

```
string s;  
s.resize(10);  
cout << s << endl;
```

این دستور 10 کاراکتر NULL رو چاپ می کنه .

```
cout << s.size() << endl;
```

این دستور عدد 10 رو تو خروجی چاپ می کنه .

2 - پیاده سازی دوم ، علاوه بر طول مقداری رو به عنوان مقدار اولیه تو هر خونه از رشته قرار میده .

```
string s;  
s.resize( 10 , 'a' );
```

این دستور طول رشته s رو برابر 10 قرار میده و همچنین کاراکتر 'a' رو تو همه خونه کپی می کنه .

```
cout << s << endl;
```

این دستور رشته aaaaaaaaaa رو چاپ می کنه .

## substr()

این تابع قسمتی از رشته رو جدا می کنه به این صورت که اندیس محل شروع زیر رشته جدید و طول زیر رشته رو می گیره و رشته ای شامل ، زیر رشته مورد نظر بر می گردونه ، این تابع به مقدار رشته اصلی اصیبی وارد نمی کنه .  
طول زیر رشته دلخواه بوده و در صورت اعلام نکردن طول ، سیستم به طور پیش فرض پایان رشته رو در نظر میگیره .

```
string s = "abc123abc";  
string t = s.substr( 3 , 3 );  
cout << t << endl;
```

این دستور رشته 123 رو چاپ می کنه .

```
string t = s.substr( 3 );  
cout << t << endl;
```

این دستور رشته 123abc رو چاپ می کنه .

### find()

این تابع رشته ای را در رشته فعلی جستجو می کند ، در صورت پیدا کردن ، اندیس اولین نقطه ای که رشته مورد نظر تکرار شده را بر می گرداند و در غیر این صورت مقدار npos رو بر می گرداند .

```
string s = "abcdefedcba";  
cout << s.find( "efe" ) << endl;
```

این دستور عدد 4 را چاپ می کند .

```
cout << s.find( "xxxx" ) << endl;
```

این دستور عدد مقدار npos را چاپ می کند .

### insert()

این تابع رشته را به رشته اصلی اضافه می کند .

برای این تابع چند نوع پیاده سازی وجود دارد که به ترتیب معرفی می کنیم .

1 - در این پیاده سازی تابع دو ورودی می گیرد ، اولی اندیسی که باید رشته در آن محل اضافه شود و دومی رشته که می خواهیم اضافه کنیم .

```
string s = "123789";  
string t = "456";  
s.insert(3 , t );  
cout << s << endl;
```

این دستور رشته 123456789 را چاپ می کند .

2 - در این پیاده سازی زیر رشته ای از یک رشته را به رشته اصلی اضافه خواهیم کرد .

این تابع 4 ورودی دارد ، اولی اندیس محل اضافه شدن و دومی رشته ای که می خواهیم اضافه شود و سومین ، اندیسی از رشته دوم به عنوان نقطه شروع زیر رشته و چهارمین طول زیر رشته است .

```
string a = "abcghu";  
string b = "aabcdef";  
a.insert( 3 , b , 2 , 3 );  
cout << a << endl;
```

این دستور رشته abcdefghi را چاپ می کند .

3 - در این پیاده سازی زیر رشته ای با شروع از اول از یک رشته را به رشته اصلی اضافه خواهیم کرد .

این تابع 3 ورودی دارد ، اولی اندیس محل اضافه شدن و دومی رشته دوم و سومی تعداد کاراکتر هایی که می خواهیم از اول رشته دوم به رشته اول اضافه شود .

```
string a = "abcghi";  
string b = "defabcabc";  
a.insert( 3 , b , 3 );  
cout << a << endl;
```

این دستور رشته abcdefghi را چاپ می کند .

این پیاده معادل قبلی است با این شرط که سومین ورودی صفر باشد .

4 - در این پیاده سازی می توانیم یک رشته از نوع رشته های زبان C را به رشته اضافه کنیم .

```
string s = "123789";  
s.insert( 3 , "456" );  
cout <<s << endl;
```

این دستور رشته 123456789 را چاپ می کند .

چند پیاده سازی دیگر هم برای insert وجود دارد که زیاد استفاده نمی شود .

## erase()

این تابع مقداری از رشته را پاک می کند . 3 نوع پیاده سازی برای این تابع داریم :

1 - این پیاده سازی ورودی نمی گیرد ؛ در این حالت تمام رشته پاک می شود . این دستور معادل clear عمل می کند .

```
string s = "123";  
s.erase();  
cout << s.size() << endl;
```

این دستور عدد 0 را چاپ می کند .

2 - این پیاده سازی یک اشاره گر به اندیسی از رشته را می گیرد و کاراکتر موجود در آن اندیس را حذف می کند .

```
string s = "1234";  
s.erase( s.begin() + 2 );  
cout << s << endl;
```

این دستور رشته 124 را چاپ می کند .

3 - این پیاده سازی یک بازه از رشته را حذف می کند .

این تابع دو ورودی می گیرد اولی اشاره گر به نقطه شروع و دومی اشاره گر به پایان بازه .  
در این پیاده سازی بازه شامل اندیس پایان نیست .

```
string s = "145623";  
s.erase( s.begin() + 1 , s.begin() + 4 );  
cout << s << endl;
```

این دستور رشته 123 را چاپ می کند .

توابع دیگری در این کتابخانه داریم اما به دلیل اینکه معمولاً استفاده نمیشن ، اینجا مطرح نمی کنیم .  
برای کسب اطلاعات بیشتر می تونید به سایت زیر مراجعه کنید .

<http://www.cplusplus.com/reference/string/string/>