

STL 기초

String

한컴에듀케이션



Strings library

3개의 string type을 제공하는 library

`basic_string`

- `string` : `basic_string<char>`
- `wstring` : `basic_string<wchar_t>`
- etc

`basic_string_view(c++17)`

- `string` : `basic_string_view<char>`
- `wstring` : `basic_string_view<wchar_t>`
- etc

Null-terminated strings

- null-terminated byte strings
- null-terminated multibyte strings
- null-terminated wide strings

string

basic_string<char>

- header : `#include<string>`
- cin, cout으로만 입출력 가능
- STL Container에서는 string으로 문자열 처리
- Random-access iterator 제공
- 전반적인 사용법이 vector와 비슷
- basic_string<char>을 typedef로 선언한 타입

```
template<
    class CharT,
    class Traits = std::char_traits<CharT>,
    class Allocator = std::allocator<CharT>
> class basic_string;
```

string 주요 문법

`iterator` str.begin(), str.end()

`reverse_iterator` str.rbegin(), str.rend()

`char` front(), back()

`bool` empty()

`ull` size()

`void` clear()

`void` push_back()

`void` pop_back()

`char` cstr[100] = "abcdefg"

`string` str = cstr : *char => string

`char*` str.c_str() : string => *char

`char` str[pos] : pos 위치의 문자 반환

* ull : unsigned long long

* 기본적인 연산자 모두 정의되어 있음

str == str2 : 문자열 일치 여부

str < str2 : str이 사전순으로 빠르면 true 반환

str += str2 : str 뒤에 str2 추가

`string` str.insert(ull index, ull count, char ch) : index 위치에 count 개의 ch 등록

`string` str.insert(ull index, char *s) : index 위치에 s 전체 등록

`string` str.insert(ull index, char *s, int count) : index 위치에 s count개만 등록

`string` str.insert(ull index, string str) : index 위치에 str 등록

`string` str.insert(ull index, string str, ull start, ull count = npos) : index 위치에 str, start 위치부터 count개 등록, count 설정 안하면 끝까지 등록

`iterator` str.insert(iterator pos, char ch) : pos 위치에 ch 등록

`string` str.erase(ull index = 0, ull count = npos) : index 위치부터 count개 삭제, index 설정 안하면 시작부터, count 설정안하면 끝까지 삭제

`string` str.erase(iterator pos) : pos 위치 문자 삭제

`string` str.substr(ull pos, ull len) : pos 위치부터 len 길이의 substring 반환

`ull` stoi(str) : str의 처음부터 digit 문자가 아닌 첫번째 문자 이전까지를 int로 반환

string example

```
char cstr[] = "stl";
string a = cstr;           // stl
a += "Easy";              // stlEasy
a[0] = 'x';               // xtlEasy
a.erase(a.begin()+3);     // xtlasy
a.erase(1, 2);            // xasy
a.substr(1, 2);           // return : as , a 그대로
string b = "123xyz45";
stoi(b);                  // 123 (int)
stoll(b);                 // 123 (long long)
a += b;                   // xasy123xyz45
a.c_str();                // xasy123xyz45 (char*)
```

C string library

string.h (cstring)

char* 를 처리하는 함수들이 저장된 library

```
strcmp()  
strcpy()  
strlen()  
  
memcmp()  
memcpy()  
memset()
```

ctype.h (cctype)

char를 처리하는 함수들이 저장된 library

```
isalpha()  
isdigit()  
islower()  
isupper()  
  
tolower()  
toupper()
```

감사합니다

