



K-Digital Training 스마트 팩토리 4기

String

std::string

- C++ STL에서 제공하는 클래스
- 문자열을 다루는 클래스
- C에서는 char* 또는 char[] 로 사용했음
- char* 또는 char[] 과 다르게 문자열의 끝에 '\0' 이 들어가지 않으며 문자열의 길이를 동적으로 변경 가능

string 사용하기

```
#include <iostream>
#include <string>

using namespace std;

int main() {
    string str;

    cin >> str; // 공백 이전까지 문자열 입력

    cout << "str1 : " << str << endl;

    getline(cin, str); // 엔터 처리
    getline(cin, str); // \n 까지 문자열 입력

    cout << "str2 : " << str << endl;

    getline(cin, str, 'a'); // 'a' 까지 문자열 입력

    cout << "str3 : " << str << endl;

    return 0;
}
```

결과

```
qwer asdf
str1 : qwer
qwer asdf
str2 : qwer asdf
qwer asdf
str3 : qwer
```

string 생성

```
#include <iostream>
#include <string>

using namespace std;

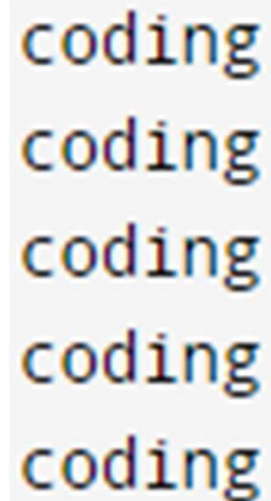
int main() {
    string str1; // 빈 문자열 생성
    string str2 = "coding";
    string str3("coding");
    string str4(str3);

    char s[] = {'c','o','d','i','n','g','\0'};
    string str5(s);
    string *str6 = new string("coding");

    cout << str1 << endl;
    cout << str2 << endl;
    cout << str3 << endl;
    cout << str4 << endl;
    cout << str5 << endl;
    cout << *str6 << endl;

    return 0;
}
```

결과



coding
coding
coding
coding
coding

string 연산

```
int main() {  
    string str1 = "coding";  
    string str2 = "on";  
  
    // 사전 순 비교  
    cout << (str1 < str2) << endl;  
  
    string str3 = "bbbbbb";  
    string str4 = "bb";  
    string str5 = "bba";  
  
    //동일할경우에 더 긴게 크다  
    cout << (str3 > str4) << endl;  
    cout << (str4 < str5) << endl;  
  
    cout << (str1 + str2) << endl;  
  
    return 0;  
}
```

결과

```
1  
1  
1  
codingon
```

string 함수

- .at(index) – index위치의 문자 반환, 범위 체크○
- [index] – index위치의 문자 반환, 범위 체크X
- .front() – 가장 앞 문자 반환
- .back() – 가장 뒤 문자 반환
- .length(), .size() – 문자열 길이
- .resize(n) – n의 크기로 변환, 기존 길이보다 작으면 남은 부분 삭제, 크면 null로 채움
- .resize(n, 'a') – 채울때 'a'로 채움
- .empty() – 빈 문자열인지 확인

string 함수

- `.append(str2)` – str2 문자열 이어 붙여줌 (+와 동일)
- `.append(str2, n, m)` – str2 문자열의 n index부터 m개의 문자 붙여줌
- `.insert(n, str2)` – n index에 str2 문자열 삽입
- `.replace(n,k,str2)` – n index부터 k개의 문자를 str2로 대체
- `.clear()` – 문자열 모두 지움
- `.erase(n, m)` – n index부터 m개의 문자열을 지움

string 함수

- `.find(str2)` – `str2`가 포함되어있는지 확인, 찾으면 첫번째 index 반환
- `.find(str2, n)` – `n` index이후부터 find
- `.substr(n)` – `n` index부터 끝까지의 문자를 부분문자열로 반환
- `.substr(n,k)` – `n` index부터 `k`개의 문자열 부분문자열로 반환
- `.compare(str2)` – 문자열 비교, 같으면 0, `str < str2` 면 음수, `str > str2` 면 양수 반환

유용한 string 함수

- `isdigit(c)` – `c`가 숫자이면 `true`, 아니면 `false`
- `isalpha(c)` – `c`가 영어이면 `true`, 아니면 `false`
- `toupper(c)` – `c`를 대문자로 변환
- `tolower(c)` – `c`를 대문자로 변환
- `stoi(str)` – `str`을 `int`로 변경
- `stof(str)` – `str`을 `float`로 변경
- `to_string(n)` – `n`을 `string`으로 변경

string 함수

```
int main() {  
    string str = "abcdefgh";  
    cout << str.at(2) << '\n';  
    cout << str[2] << '\n';  
    cout << str.front() << '\n';  
    cout << str.back() << '\n';  
    cout << str.length() << '\n';  
    str.resize(6);  
    cout << str << '\n';  
    cout << str.empty() << '\n';  
    str.clear();  
    cout << str.empty() << '\n';  
  
    return 0;  
}
```

결과

```
c  
c  
a  
h  
8  
abcdef  
0  
1
```

string 함수

```
int main() {  
    string str = "coding";  
    string str2 = "on";  
    str.append(str2, 1, 1);  
    cout << str << '\n';  
  
    str = "coding";  
    str.insert(2, str2);  
    cout << str << '\n';  
  
    str = "coding";  
    str.replace(1, 2, "zzz");  
    cout << str << '\n';  
    str.erase(0, 2);  
    cout << str << '\n';  
  
    return 0;  
}
```

결과

```
codingn  
coonding  
czzzing  
zzing
```

string 함수

```
int main() {  
    string str = "coding";  
    string str2 = "on";  
    cout << str.find("od") << '\n';  
    cout << str.substr(1, 3) << '\n';  
    cout << str.compare(str2) << '\n';  
    cout << str.compare("coding") << '\n';  
  
    return 0;  
}
```

결과

```
1  
odi  
-786175  
0
```

string 함수

```
int main() {  
    cout << isdigit('1') << endl;  
    cout << isdigit('a') << endl;  
    cout << isalpha('1') << endl;  
    cout << isalpha('a') << endl << endl;  
  
    cout << toupper('a') << endl;  
    cout << tolower('A') << endl << endl;  
  
    cout << char(toupper('a')) << endl;  
    cout << char(tolower('A')) << endl << endl;  
  
    int a = stoi("1234");  
    float b = stof("12.34");  
    string c = to_string(1234);  
    cout << a << endl;  
    cout << b << endl;  
    cout << c << endl;  
  
    return 0;  
}
```

결과

```
1  
0  
0  
1024  
  
65  
97  
  
A  
a  
  
1234  
12.34  
1234
```

실습1 string 사용해보기

1. string s = "Police say two people are suspected of trying to create a shortcut for their construction work.The two have been detained and the case is under further investigation.The 38-year-old man and 55-year-old woman were working near the affected area, the 32nd Great Wall.";
2. s 문자열의 길이 출력
3. 100번째 문자 출력(index는 0부터 시작)
4. "two" 라는 문자가 처음 나오는 index 출력
5. "two" 라는 문자가 두번째 나오는 index 출력

265

w

11

99

실습2 string 사용해보기

1. 두 문자열을 입력 받아서 둘 모두 숫자인지 검사, 아니면 다시 입력 받도록
 1. "1111" -> OK
 2. "abcd", "a123", "123a" -> NO
2. 앞에서 입력 받은 두 숫자를 이어 붙여서 출력
 1. "1111" "2222" -> "11112222"
3. 앞에서 입력 받은 두 숫자의 합을 출력
 1. "1111" "2222" -> 3333

실습3 string 사용해보기

1. `string s = "Codingon"`
2. 첫번째 문자 소문자로 변경 -> `"codingon"`
3. `"ding"` 이라는 부분문자열 반환
4. `"cooooooon"` 이 되도록 변경
5. `"con"` 이 되도록 변경