



lesson 8 string

في لغة **C** عندما كنا نريد حجز اسم كنا نقوم بحجز **Array of chars** لكن في لغة **C++** عندما نقوم بحجز اسم نقوم بتضمين مكتبة تسمى **string** ويجب التفريق بين هذه المكتبة ومكتبة **string.h** في لغة **C**

وبالتالي **string** في لغة **C++** هي **data type** مثلها مثل **char** - **float** - **int** ثم من مميزات **String** في **C++** أننا لا نقوم بحجز عدد معين له كما كنا نفعل مع **array of chars** لكن نقوم بحجز **string** وهو يقوم بالحجز تلقائياً على حجمه

```
string x = "Gammal Tech";
```

مثال :

```
#include <iostream>
#include <string>
using namespace std;
```

```
int main() {
    string x = "Gammal Tech";
    cout << x;
}
```

output:

Gammal Tech

إذا أردنا معرفة عدد حروف الكلام المسجل داخل **string** نقوم باستخدام **size()** سنقوم بعد كل الموجود داخل علامة التنصيص " " بما فيهم المسافة

```
x.size();
```

مثال :

```
string x = "Gammal Tech",y;
cout << y.size() << " " << x.size();
output:
```



0 11

كما يمكننا نقل **string** ممثلي إلى **string** آخر كما فعلنا في **vector**

```
string x = "Gammal Tech",y;
```

```
y = x;
```

```
cout<<y;
```

output:

Gammal Tech

وبهذا نكون نقلنا كل مافي **x** إلى **y** بخطوة واحدة وسيتم مسح الكلام الموجودة في **y** إذا وجد

كما يمكننا أيضاً في لغة **C++** حذف مافي **string** وجعله فارغاً
عن طريق خطوة واحدة باستخدام **clear()**;

```
string x = "Gammal Tech";
```

```
x.clear();
```

```
cout<<x;
```

output:

لن يقوم بطباعة شيء لأننا قمنا بمسح كل شيء داخل **string**

ماذا إذا أردنا مقارنة **string** مع **string** آخر
كنا في لغة **C** نقوم بتضمين مكتبة **string.h** واستخدام أمر **strcmp()**؛
لكن في لغة **C++** يمكننا ببساطة فعل ذلك عن طريق مقارنتهم ببعض

```
if (x == y)
```



مثال :

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    string x, y;
    x = "Gammal Tech";
    y = x;
    if (x == y)
        cout << "Equal";
}
```

output:

Equal



كما يمكننى عمل **vector** أو **Set** من **string**
مثال :

إدخال **strings** داخل **vector** وطباعة حجمه وطباعته بالترتيب :

```
#include <iostream>
#include <vector>
#include <string>
#include <algorithm>
using namespace std;

int main() {
    vector <string> x;
    x.push_back("Gammal Tech");
    x.push_back("www.gammal.tech");
    x.push_back("C++ Programming");
    x.push_back("Gammal Tech");
    sort(x.begin(), x.end());
    cout << x.size()<<endl;
    for (int i = 0; i < x.size(); i++)
        cout << x[i]<<endl;
}
```

output:

4

C++ Programming



Gammal Tech
Gammal Tech
www.gammal.tech

كما يمكننا استخدام **strings** مع **set** أيضاً وستقوم بترتيبهم تلقائياً
مثال :

```
#include <iostream>
#include <set>
#include <string>
#include <algorithm>
using namespace std;
int main() {
    set <string> x;
    x.insert("Gammal Tech");
    x.insert("www.gammal.tech");
    x.insert("C++ Programming");
    x.insert("Gammal Tech");
    cout << x.size() << endl;
    set<string> :: iterator it;
    for (it = x.begin(); it != x.end(); it++)
        cout << (*it) << endl;
}
```

output:

3
C++ Programming
Gammal Tech
www.gammal.tech



هنا سيقوم بطباعة حجم **set** وعناصرها مرتبين ويقوم بحذف العنصر المكرر