

lesson 5 CPP Vector

فى لغة **C** كان هناك **Array**، كنا نقوم بحجزه ب عدد معين ، فيظل بهذا العدد إلى نهاية البرنامج ، لكن فى **C++** هناك مايسمى **Vector** مثله مثل **Array** لكن ميزته انه يمكننى التحكم فى مساحته، استطيع في بداية البرنامج ان اقوم بحجزه **0** وفى منتصف البرنامج اجعله **100** وبعد فترة اجعله ب **10**، يمكننى التحكم فى مساحته أثناء كتابة الكود

ونقوم بتضمينه من خلال مكتبة تسمى **vector** وتعريفه عن طريق كتابة **vector** ثم يليه **< >** وبداخلها نوع المتغير ثم اسم **vector**

```
vector <data type> name;
```

ونقوم بتعريفه بهذه الطريقة :

```
#include <vector>
int main() {
    vector <int> x;
}
```

فى هذه اللحظة **vector** ليس به اى قيم ، ونضيف إليه القيم يكون باستخدام أمر **push.back()**

مثال :

إذا أردنا إضافة العناصر التالية إلى **vector**

5 , 9 , 200 , -3

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

```
int main() {
    vector <int> x;
    x.push_back(5);
    x.push_back(9);
    x.push_back(200);
    x.push_back(-3);
}
```

وهنا **vector** مثل **array** فى ترتيب العناصر :

```
x[0] = 5
x[1] = 9
x[2] = 200
x[3] = -3
```

ويمكنك طباعة العنصر الذي تريده بهذه الطريقة :

```
cout << x[0] << endl;
```

output :

5

```
cout << x[3] << endl;
```

output :

-3

ولطباعة عناصر **vector** سيكون بهذه الطريقة :

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

int main() {
    vector<int> x;
    x.push_back(5);
    x.push_back(9);
    x.push_back(200);
    x.push_back(-3);
    for (int i = 0; i < 4; i++)
        cout << x[i] << endl;
}
```

output :

```
5
9
200
-3
```

وهنا كنا نعرف عددهم، لكن ماذا إذا ما كنا نعرف عددهم ؟
نستخدم **size()** وهو يقوم بعدها تلقائياً :

```
for (int i = 0; i < x.size(); i++)
    cout << x[i] << endl;
```

كنا ذكرنا أن من مميزات **vector** يمكنني تغيير حجمه أثناء عمل البرنامج
مثال :

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

int main() {
    vector<int> x;
    cout << x.size() << endl;
    x.push_back(5);
    cout << x.size() << endl;
    x.push_back(9);
    x.push_back(200);
    cout << x.size() << endl;
    x.push_back(-3);
    cout << x.size() << endl;
}
```

output :

0
1
3
4

وبالتالي شاهدنا حجم **vector** في مراحل مختلفة

ومن مميزات **Vector** أن انقل من **vector** ممتلئ إلى **vector** فارغ
مثال :

vector x يحتوى 4 عناصر و **vector y** فارغ ، نريد نقل العناصر ل **y** أيضاً
y = x;
بهذه الطريقة قمنا بنقل كل مافى **x** إلى **y** ونقوم بطباعتها للتأكد كالتالى :

```
#include <iostream>
```

```
#include <vector>
```

```
using namespace std;
```

```
int main() {  
    vector <int> x;  
    vector <int> y;  
    x.push_back(5);  
    x.push_back(9);  
    x.push_back(200);  
    x.push_back(-3);  
    y = x;  
    for (int i = 0; i < y.size(); i++)  
        cout << y[i] << endl;  
}
```

output :

5

9

200

-3

كما يمكننى مسح العناصر الموجودة داخل **vector** من خلال أمر **clear ()**
مثال :

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

int main() {
    vector <int> x;
    vector <int> y;
    x.push_back(5);
    x.push_back(9);
    x.push_back(200);
    x.push_back(-3);
    x.clear();
    cout << x.size() << endl;
}
```

output:

0

كما قلنا أنه يمكننا فى **vector** تكبير حجمه أو تصغيره ،إثناء العمل وحذف العناصر الموجودة به ونقل العناصر **vector** آخر