Phill-C++3

陣列

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

int main()
{
    int a[3]={10,20,30};
    cout << a[0]<<endl; //rlvalue ->取值

    a[2]=87; // lvalue -> 賦值

    cout << a[2]<<endl; //垃圾
    return 0;
}
```

字串在C就是字元陣列

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

int main()
{
    char a[3]={'a','b','c'};
    cout << a[0]<<endl; //取值

    a[2]='z'; //賦值

    cout << a[2]<<endl; //垃圾
    return 0;
}
```

ASCII

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

int main()
{
    char a[]={'a','b','c'};
    printf("%d\n", a[0]);
    printf("%c\n", a[0]);
    return 0;
}
```

C-style 字串

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

int main()
{
    const char *a="abc";

    int length =  strlen(a);
    cout << "length=" << length;
    return 0;
}
```

拷貝字串

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

int main()
{
    char arr1[]= "hello world";
    char arr2[87];

    //string copy
    strcpy(arr2, arr1);
    cout << arr2<< endl;
    return 0;
}
```

限制長度的拷貝

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

int main()
{
    char arr1[]= "hhhhhhhhhhhhhhhhhhhhhhhhhh";
    char arr2[5];
    //string copy
    strncpy(arr2, arr1, sizeof(arr2)-1); //安全
    arr2[sizeof(arr2)]='\0';
    cout << arr2<< endl;
    return 0;
}
```

sizeof vs. strlen

- sizeof

  c++→ not function → compiling → .exe 就已經知道大小了→ compiling time → int structure, float array → 總結構的大小

- strlen

  function → runtime → char array only → 不包含 \0

字串疊加

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

int main()
{
    char arr1[]= "hello";
    char arr2[]= "world";
    char arr3[256];

    strcpy(arr3, arr1);
    strcat(arr3, arr2);
    cout << arr3 <<endl;
    return 0;
}
```

C++ 字串

string 類別

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

int main()
{
    string my_string1="hello";
    string my_string2=" world";

    cout << my_string1.length(); // string method
    cout << my_string1+my_string2;
    return 0;
}
```

C→ C++

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

int main()
{
    //Cstyle -> C++ class
    char char_array[]="Phill is good.";
    string str(char_array);

    cout << str << endl;
    return 0;
}
```

c++→C

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

int main()
{
    string str ="Phill is good.";
    const char* char_array= str.c_str();
    cout << char_array << endl;

    // error!!
    char_array[3]='z';
    return 0;
}
```

作業

C-style → function :大寫的string