



lesson 78 sprintf

فى هذا الدرس سندرس أمر **sprintf** لكن لماذا نقوم بدراسة هذا الأمر فى التمارين (**practice**)، لأن مهارة البرمجة هي ال **practice** وعمل برامج مختلفة الأفكار، إذا كنت تعلم عدد كبير من الأوامر لكن لا تستطيع عمل برنامج، هكذا لن تكون مبرمج متميز، لكن إذا كنت تعلم عدد قليل من الأوامر وتقوم بعمل برامج حتى إذا كان عدد سطور البرنامج كثيرة، لكن تستطيع كتابة البرنامج هكذا أنت متميز و ذكى و تستطيع إيجاد الحلول .

أهم شئ هي طريقة التعامل مع الأوامر وليس مقدار حفظك للأوامر.

و الآن هيا نعرف فى إى شئ نستخدم أمر **sprintf** !؟

يستخدم هذا الأمر إذا قمنا بتعريف **array of characters** لكن لم نضع فيه اسم ونريد بعد عدد معين من الأكواد اضافة كلام لتسجيل كلام فى هذه **Array** مثال :

```
#include <stdio.h>
```

```
int main( ) {  
    char x[100] = "Programming is fun";  
    printf("%s\n", x);  
}
```

output:

Programming is fun

(قم بتجربة الكود بنفسك واضغط هنا)



هنا قام بالطباعة بصورة طبيعية لأننا قمنا بتسجيل الكلام في **array of characters**،
لكن ماذا إذا لم نسجله؟
ونحتاج أن نسجله فيما بعد كالتالي :

```
#include <stdio.h>
```

```
int main( ) {
    char x[100];
    x = "Programming is fun";
    // سيظهر هنا خطأ
    printf("%s\n", x);
}
```

(قم بتجربة الكود بنفسك واضغط هنا)

لكن كيف يمكننا تصحيح هذا الخطأ، هناك أمر درسنا سابقاً وهو **strcat** وسيكون
البرنامج كالتالي :

```
#include <stdio.h>
#include <string.h>
```

```
int main( ) {
    char x[100];
    x[0] = '\0';
    // بهذا الأمر نجعل قيمة x فارغة وغير موجود بها أى شئ
    // x[0] = 0;
    // أو بتلك الطريقة
    strcat(x, "Programming is fun");
    // هنا قمنا بنقل الكلام إلى x الفارغة
    printf("%s\n", x);
}
```

output:



Programming is fun

(قم بتجربة الكود بنفسك واضغط هنا)

هنا عمل بصورة صحيحة، ولذلك كلما كان بإمكان عمل البرنامج المراد عمله وانت تعرف أوامر قليلة، لكن تستطيع عمل البرنامج فهذا هو المراد

لكن الفرق في الأمر الجديد أننا يمكننا اختصار السطور وسيكون استخدام أمر **sprintf** كالتالي :

نقوم بكتابة أمر **sprintf** ثم () ثم اسم ال **string** الذي سنقوم بنقل الكلام فيه ثم فاصلة , ثم " " وبداخلهم الكلام المراد كتابته

```
#include <stdio.h>
```

```
int main( ) {
```

```
    char x[100];
```

```
    sprintf(x, "Programming is fun");
```

```
    // نقوم بإضافة النص إلى x
```

```
    printf("%s\n", x);
```

```
}
```

output:

Programming is fun

(قم بتجربة الكود بنفسك واضغط هنا)

أمر **printf** يقوم بالطباعة على الشاشة

أمر **fprintf** يقوم بالطباعة داخل ملف

أمر **sprintf** يقوم بالطباعة داخل **string**

لكن هناك فرق آخر وهو أننا في أمر **sprintf** إذا استخدمناه لنفس ال **string** أكثر من مرة فهو يقوم بمسح الكلام المسجل وإضافة الكلام الجديد المراد إضافته



مثال :

```
#include <stdio.h>

int main( ) {
    char x[100];
    sprintf(x, "Programming is fun");
    sprintf(x, "Hello Gammal Tech");
    // سيقوم بمسح الكلام الموجود داخل x قبل هذا الرقم
    printf("%s\n", x);
}
```

output:

Hello Gammal Tech

(قم بتجربة الكود بنفسك واضغط هنا)

ولذلك نحن لسنا بحاجة إلى كتابة $x[0] = 0$ ، لأنه تلقائياً يقوم بمسح الكلام المسجل قبل تنفيذ الأمر