



Lesson 2 Variables

الدرس السابق تعلمنا كيفية جعل الكمبيوتر يتكلم باستخدام أمر **printf**. في هذا الدرس سنتعرف على كيفية تخزين الكمبيوتر للبيانات. للقيام بذلك ، يستخدم الكمبيوتر ما يسمى بال (متغيرات أو **variables**) باختصار ، ال **variables** هي مجرد حاويات للبيانات التي تريد تخزينها ، سواء كان ذلك رقماً أو حرف أو كلمة

لإنشاء متغير يخزن عدداً صحيحاً (عدد بدون كسور)، ما عليك سوى كتابة ما يلي:

int اسم المتغير ;

على سبيل المثال :

int x ;

كلمة **int** هي نوع المتغير (data type) و هي اختصار لكلمة **integer** و **x** هو الاسم الذي اعطيناه للمتغير حالياً .

المتغير الذي أنشأناه للتو - أي **x** - لا يحتوي على أي بيانات مخزنة بداخله ، لأننا لم نضف أي قيمة أو "بيانات" ليتم تصنيفها على أنها "البيانات المخزنة في المتغير **x**" إذن كيف نضيف قيمة داخل متغير؟

هذا سهل! كل ما عليك القيام به هو إضافة علامة يساوي (=) وأي بيانات ترغب في تخزينها في المتغير (مع التأكد من أنها من نفس النوع الذي يستقبله المتغير) ستكون على النحو التالي:

int x = 5 ;

الآن المتغير **x** له قيمة (5) مخزنة بداخله

يمكنك إنشاء متغيرات متعددة، يمكنك أيضاً إضافة أو طرح أو إجراء عمليات حسابية أخرى على متغيرات عدد صحيح مثل

int x = 5 ;

int y = 3 ;

int output ;

output = y+x ;



لاحظ في هذا المثال قمنا بحجز متغير اسمه (output) و لم ندخل فيه قيمة اثناء تعريفه و لكن قمنا بإدخال القيمة فيما بعد .

يمكنك أيضًا طباعة الناتج !

إليك كيفية طباعة متغير من نوع **int** :

printf ("%d" , اسم المتغير);

علامة **%d** تسمى Format Specifier و هي ما يحدد نوع البيانات التي سيتم طباعتها على الشاشة، لذلك **%d** تعني ان الذي يتم طباعته هو رقم صحيح (int)

بعد ال double quotes " " يتم وضع علامة (,) و بعدها يتم وضع اسم المتغير الذي يتم طباعة قيمته مكان ال **%d**

لذلك لطباعة متغير ال **output** " نكتب ببساطة:

printf ("%d" , output) ;

الناتج : 8

تعلمنا أحد أنواع ال **data types** و هو ال **int** الذي يعبر عن العدد الصحيح، إذا ما هي الأنواع الأخرى من ال **data types** ؟

ال **data types** للأرقام الصحيحة

short , int , long , long long

ال **data types** للأرقام العشرية

float , double

ال **data types** للرموز و الحروف الأبجدية

char



Data Type	Size (bytes)	Format Specifier	نوع البيانات المُخزنة
short	2 (bytes)	%hd	عدد صحيح
int	2 or 4 (bytes)	%d	عدد صحيح
long	4 or 8 (bytes)	%ld	عدد صحيح
long long	8 (bytes)	%lld	عدد صحيح
float	4 (bytes)	%f	عدد عشري
double	8 (bytes)	%lf	عدد عشري
char	1 (byte)	%c	حرف او رمز

ال size يختلف من اصدار إلى اخر من أنظمة التشغيل (operating systems)

ليس عليك حفظ هذا الجدول، لكن كل ما يجب معرفته حالياً هو نوع البيانات التي يخزنها كل data type .

الانواع التي سنستخدمها بشكل أكثر خلال دراستنا هي int - float - char

```
int x = 5;
```

```
float y = 7.3;
```

```
char z = 'A';
```

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

```
printf ("%f \n",y);
```

```
printf ("%c \n",z);
```

النتائج :

5

7.3000

A



كان يمكننا أن نقوم بعمل أمر طباعة واحد هكذا :

```
printf ("%d \n %f \n %c \n",x,y,z);
```

هل لاحظت الآتي :

- عند استخدام char قمنا بوضع علامتي ' ' (single quotes) و وضعنا بينهم

الحرف 'A'

- عند طباعة عدد من نوع float يتم طباعة عدد من الأصفار بعد العلامة العشرية،

إذا أردت التحكم في عدد الأرقام المطوعة بعد العلامة العشرية نكتب %0.1f بدل

من %f ، و يمكننا تغيير 0.1 كيفما شئنا 0.0 , 0.1 , 0.2 , 0.3 ...

الأخطاء الشائعة في الكود:

- استخدام متغير قبل التصريح عنه
- استخدام متغير **int** حيث يكون البيانات المدخلة من نوع **float** أكثر ملائمة
- نسيان الفاصلة المنقوطة " ; "