



## lesson 25 ASCII

يرى الكمبيوتر كل شئ على هيئة أرقام حتى تلك الحروف الانجليزية (a,b,c) حتى أشكال الأرقام كما درسنا فى الدروس السابقة

### decimal, hexadecimal, octal, binary

فإن أشكال الأرقام فى هذه الانظمة، هو بالنسبة للكمبيوتر شكل هذه الأرقام مخزن ايضاً على هيئة أرقام أخرى غير الذى نعرفه  
مثلاً رقم 0 بالنسبة للكمبيوتر شكل الصفر هو مسجل برقم آخر غير **الصفر** و هو 48

لذلك عندما كنا نقوم بتعريف وحجز متغير (**variable**)  
بحرف **small** كنا نستخدم نفس الحرف **small** فى كل الكود

`int x;`

هنا **x** حرف **small**، وهذا معناه أن شكل حرف **x** بالنسبة للكمبيوتر له رقم معين،  
وشكل **X** ال **capital** بالنسبة للكمبيوتر له رقم مختلف تمام، وإذا قمنا أثناء كتابة الكود  
بكتابة **X capital** فى أى سطر برمجى سيظهر لنا خطأ لأن قيمة **small x** مختلفة  
عن **X capital** بالنسبة للكمبيوتر **small** مسجل برقم، و ال **capital** مسجل برقم  
مختلف آخر

فى هذا الدرس سنتعرف على الارقام المخزنة للأشكال من الحروف والارقام

**ASCII** هو إختصار لجملة : " American Standard Code for Information Interchange " و هو مجموعة رموز ونظام ترميز مبني على  
الابجدية اللاتينية بالشكل الذى تستخدم به فى الإنجليزية الحديثة ولغات غرب أوروبية  
أخرى.



و عندما نقول ال ASCII الخاص بالحرف كذا، هذا معناه أننا نريد أن نعرف الرقم المخزن لهذا الحرف أو هذا " الشكل " .

```
int main( ) {
    int s = 'A';
    printf(" %d ", s);
}
```

output:

65

إذا قمنا بكتابة الحرف بمفرده ، هذا معناه انه عبارة عن متغير (variable) أما إذا كتبناه كما الحال بالنسبة لحرف A داخل quote ' ' هنا يعتبر شكل الحرف وهو مخزن داخل الكمبيوتر برقم ، وعند قيام الكمبيوتر بطباعته فهو غير مخزن كشكل وإنما مخزن رقم، فيذهب البرنامج لينظر إلى ال ASCII ويرى أن هذا الشكل مُخزن برقم ماذا؟ ويقوم بطباعة الشكل بناء على رقم ال ASCII الخاص به

يمكنك طباعة الرقم المخزن لشكل A بهذه الطريقة

```
int main( ) {
    printf(" %d ", 'A');
}
```

output :

65

هنا وضعنا quote ووضعنا شكل الحرف المراد طباعة الرقم المخزن به.



**تذكر** في الدرس الثاني ( variables ) ذكرنا أن لطباعة الحرف أو الشكل أو الرمز نستخدم %c و الذي عادة ما نستخدمه عندما نطبع قيمة متغير من نوع char ( سندرسه بالتفصيل فيما بعد ) لكن يمكننا أيضاً أن نطلع الشكل باستخدام %c حتى و لو كان مُخزن في متغير من نوع int .

إذا أردنا طباعة أشكال الحروف capital والأرقام المخزنة بها سنقوم بعملها بهذه الطريقة :

```
int main( ) {
    int i;
    for (i = 'A'; i <= 'Z'; i++)
        printf(" %c %d ", i, i);
}
```

**output :**

```
A 65 B 66 C 67 D 68 E 69 F 70 G 71 H 72
I 73 J 74 K 75 L 76 M 77 N 78 O 79 P 80 Q
81 R 82 S 83 T 84 U 85 V 86 W 87 X 88 Y 89 Z
90
```

( قم بتجربة الكود بنفسك واضغط هنا )

كذلك الأمر بالنسبة للحروف small  
نقوم بطباعتها بهذه الطريقة

```
int main( ) {
    int i;
    for (i = 'a'; i <= 'z'; i++)
```



```
printf(" %c %d ", i, i);
}
```

### output:

```
a 97 b 98 c 99 d 100 e 101 f 102 g 103 h 104 i
105 j 106 k 107 l 108 m 109 n 110 o 111 p 112 q
113 r 114 s 115 t 116 u 117 v 118 w 119 x 120 y
121 z 122
```

كذلك الأمر بالنسبة لأشكال الأرقام سنجد انها مخزنة على هيئة أرقام أخرى

```
int main( ) {
    int i;
    for (i = '0'; i <= '9'; i++)
        printf(" %c\t%d \n", i, i);
}
```

### output:

```
0 48
1 49
2 50
3 51
4 52
5 53
6 54
7 55
8 56
9 57
```

( قم بتجربة الكود بنفسك واضغط هنا )



لكننا قمنا بوضع 0 و 9 بين quote هكذا '9' و '0'

هنا شكل 0 مخزن على هيئة 48

شكل 1 مخزن على هيئة 49

شكل 3 مخزن على هيئة 51

وعلى هذا المنوال

الحروف والارقام فى حالة أنها شكل شئ والارقام كقيمة شئ آخر

لكن لماذا نحتاج أن نسجل الارقام كشكل لماذا ؟

بببب دائما نحتاج أن نقوم بعمل عمليات حسابية على الأرقام، فمثلا لو عمرك 18 عام

سيصبح العام القادم 19 إذا فهو رقم يزيد، لكن رقم هاتفك هل من الممكن أن يزيد العام

القادم او يمكنك اضافة رقم عليهم او جمعه او طرحه ، الاجابة لا .

إذا هنا الكمبيوتر يقوم بتخزينه على هيئة أشكال لأن لن تتم عليه اى عمليات حسابية لا

جمع او طرح او قسمة إذا يقوم بتسجيله على هيئة أشكال

مثلاً عنوانك 30 شارع محمود العقاد هل سيكون الاسبوع القادم 31 شارع العقاد لا

سيظل 30 هل بعد شهر سيصبح 35 لا سيظل 30 إذا هذا يعتبر شكل وليس رقم بالنسبة

للكمبيوتر

لذلك الارقام يمكن التعامل معها إما كقيمة أو أشكال، و فى حالة الأشكال يقوم الكمبيوتر

بتخزينها بقيمة مختلفة عن القيمة الطبيعية التى نعرفها يسجلها، بقيمة ال ASCII