

1 -

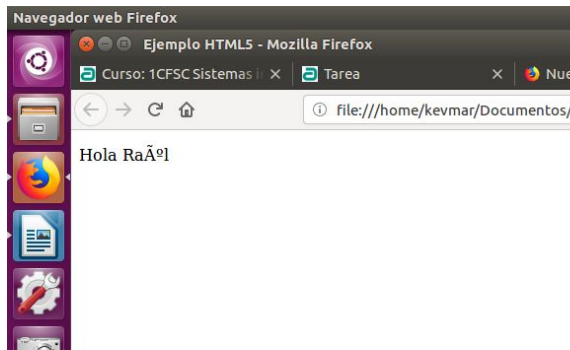


Imagen1.0

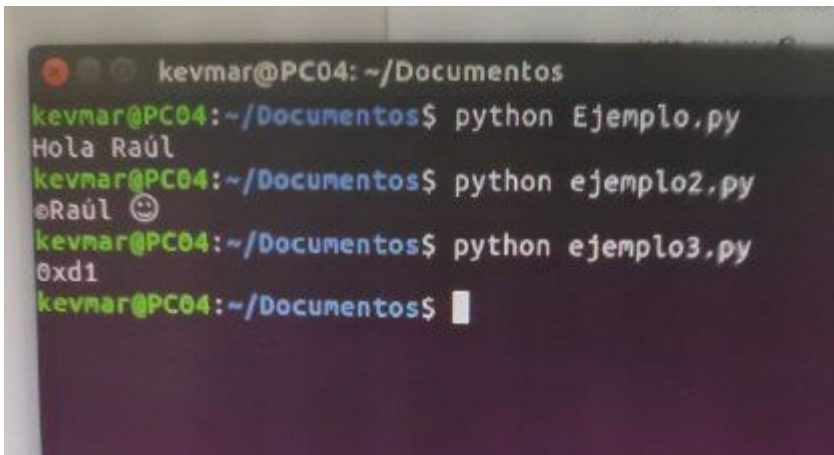


Imagen 1.1

a)

En la imagen se puede leer: Hola

b)

En la imagen 1.0 podemos ver como el acento de la palabra Raúl no está reconocida por el sistema que usamos actualmente por eso la representa de este modo.

A continuación agregando: al programa, le damos lo necesario para que pueda reconocer más caracteres y así poder leer: Hola Raúl , tal cual queremos nosotros, lo podemos ver en la Imagen 1.1.

c)

En el ejemplo2 de la Imagen 1.1 lo que muestra es Raúl pero ahora con otros caracteres, que estan escritos en la cadena. La palabra raul esta en hexadecimal

d)

Lo que se muestra en el ejemplo3 de la imagen 1.1 es la letra Ñ expresada en hexadecimal.

2-

a)

b)

c) Usar `<meta charset="UTF-8">`

3-

Usando el “Shift+Ctrl+U”

El u40 es igual a @  
El u7E es igual a ~  
El uA9 es igual a ©  
El uAE es igual a ®  
El 20AC es igual a €  
el 2030 es igual a ‰

4-

El numero de bits necesario es 5.

$$2^5 = 32$$

00000 -- 0  
00001 – 1  
00010 – 2  
00011 – 3  
00100 – a  
00101 – e  
00110 – i  
00111 – o  
01000 – u  
01001 – A  
01010 – E  
01011 – I  
01100 – O  
01101 – U  
01110 – S  
01111 – R  
10000 – P

SERPIS — 01110 01010 01111 1000 01011 01110

5-

Combinacion con 4 bits	Valor en signo-magnitud	Valor en coma fija
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	0	8
1001	-1	9
1010	-2	10
1011	-3	11
1100	-4	12
1101	-5	13
1110	-6	14
1111	-7	15

6-

Exceso 128 con 8 bits

a) 6.

$128 + 6 = 134$  en binario = 1000 0110

b) -11.

$128 + (-11) = 117$  en binario = 0111 0101

c) -2

$128 + (-2) = 126$  en binario = 0111 1110

d) 0

$128 + 0 = 128$  en binario = 1000 0000

e) 106

$128 + 106 = 234$  en binario 1110 1010

f) -20

$128 - 20 = 108$  en binario 0110 1100

g) 156

$128 + 156 = 284$  en binario con 8 bits no se puede representar, necesitamos un bit mas para poder representarlo.

## 7- En complemento a2

- a)  $101001110 == 010110010$
- b)  $101101110 == 010010010$
- c)  $101011111000 == 010100001000$
- d)  $11000110001011 == 00111001110101$
- e)  $11101100110000 == 00010011010000$

## 8 – Representa en BCD

- a) 34 — 0011 0100
- b) 217 — 0010 0001 0111
- c) 2523 — 0010 0101 0010 0011
- d) 77900 — 0111 0111 1001 0000 0000
- e) 120021 — 0001 0010 0000 0000 0010 0001

## 9- BCD Empaquetado y desempaquetado

El BCD desempaquetado completa hasta 1 Byte con 0. Sin embargo, el BCD empaquetado no completa.

Desempaquetado

146 = 00000001 00000100 00000110

Empaquetado

146: 0001010001101100

## 10- Normaliza en IEEE 754 simple precision o coma flotante.

- a)  $0 == 0\ 00000000\ 00000000000000000000000000000000$
- b)  $4,05 == 0\ 10000001\ 00000011001100110011001100110011$
- c)  $-2,25 == 1\ 10000000\ 00100000000000000000000000000000$
- d)  $-25,75 == 1\ 10000011\ 10011100000000000000000000000000$
- e)  $10,8 == 0\ 10000001\ 01110011001100110011001100110011$
- f)  $-102,2 == 1\ 10000101\ 10011000110011001100110011001100$

## 11- Resta con Ca2 y longitudes de palabra de 8 bits

- a)  $11001110 - 10110111 = 10111$
- b)  $11001001 - 1101010 = 1011111$
- c)  $1010011 - 111100 = 10111$
- d)  $11011101 - 10101 = 1101000$

12- Nos compramos un disco duro de 4 TB en S.I., pero nuestro sistema operativo trabaja con el sistema tradicional binario. ¿Qué diferencia de tamaño (entre MB y MiB) encontraremos al instalarlo en nuestro PC y comprobar el espacio disponible?

TB =  $4 \cdot 10^{12}$  bytes -- TiB =  $4 \cdot 2^{40}$  bytes Hay una diferencia del 10%.

4 TB a GB =  $4 \cdot 1.000 = 4.000$  GB. De GB a MB  $4.000 \text{ a } 1.000 \text{ MB} = 4.000.000 \text{ MB}$

4 TiB a GiB =  $4 \cdot 1.024 = 4.096$  GiB. De GiB a MiB  $4096 \cdot 1024 \text{ MiB} = 4.194.304 \text{ MiB}$

Diferencia de tamaño es  $4.194.304 - 4.000.000 = 194.304$