

I - RAPPEL (vu en seconde)

Lorsque vous appuyez sur une touche de votre clavier, un code binaire est généré :

ASCII TABLE

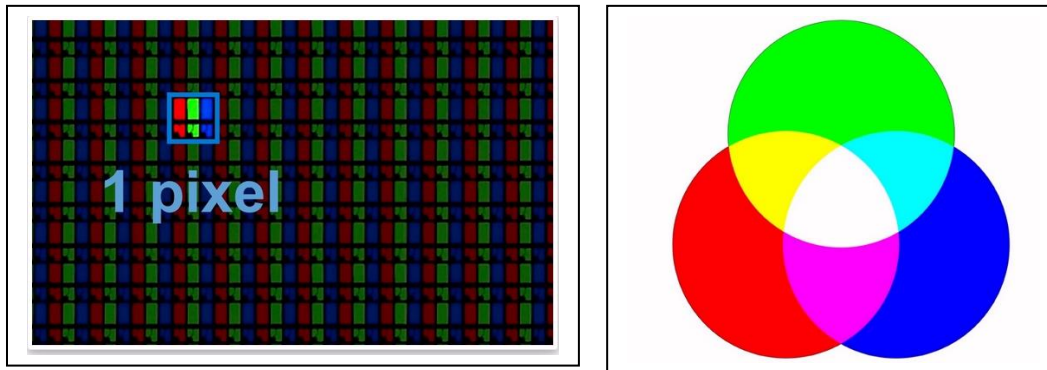
Decimal	Hexadecimal	Binary	Octal	Char	Decimal	Hexadecimal	Binary	Octal	Char	Decimal	Hexadecimal	Binary	Octal	Char
0	0	0	0	[NULL]	48	30	110000	60	0	96	60	1100000	140	`
1	1	1	1	[START OF HEADING]	49	31	110001	61	1	97	61	1100001	141	a
2	2	10	2	[START OF TEXT]	50	32	110010	62	2	98	62	1100010	142	b
3	3	11	3	[END OF TEXT]	51	33	110011	63	3	99	63	1100011	143	c
4	4	100	4	[END OF TRANSMISSION]	52	34	110100	64	4	100	64	1100100	144	d
5	5	101	5	[ENQUIRY]	53	35	110101	65	5	101	65	1100101	145	e
6	6	110	6	[ACKNOWLEDGE]	54	36	110110	66	6	102	66	1100110	146	f
7	7	111	7	[BELL]	55	37	110111	67	7	103	67	1100111	147	g
8	8	1000	10	[BACKSPACE]	56	38	111000	70	8	104	68	1101000	150	h
9	9	1001	11	[HORIZONTAL TAB]	57	39	111001	71	9	105	69	1101001	151	i
10	A	1010	12	[LINE FEED]	58	3A	111010	72	:	106	6A	1101010	152	j
11	B	1011	13	[VERTICAL TAB]	59	3B	111011	73	;	107	6B	1101011	153	k
12	C	1100	14	[FORM FEED]	60	3C	111100	74	<	108	6C	1101100	154	l
13	D	1101	15	[CARRIAGE RETURN]	61	3D	111101	75	=	109	6D	1101101	155	m
14	E	1110	16	[SHIFT OUT]	62	3E	111110	76	>	110	6E	1101110	156	n
15	F	1111	17	[SHIFT IN]	63	3F	111111	77	?	111	6F	1101111	157	o
16	10	10000	20	[DATA LINK ESCAPE]	64	40	1000000	100	@	112	70	1110000	160	p
17	11	10001	21	[DEVICE CONTROL 1]	65	41	1000001	101	A	113	71	1110001	161	q
18	12	10010	22	[DEVICE CONTROL 2]	66	42	1000010	102	B	114	72	1110010	162	r
19	13	10011	23	[DEVICE CONTROL 3]	67	43	1000011	103	C	115	73	1110011	163	s
20	14	10100	24	[DEVICE CONTROL 4]	68	44	1000100	104	D	116	74	1110100	164	t
21	15	10101	25	[NEGATIVE ACKNOWLEDGE]	69	45	1000101	105	E	117	75	1110101	165	u
22	16	10110	26	[SYNCHRONOUS IDLE]	70	46	1000110	106	F	118	76	1110110	166	v
23	17	10111	27	[ENG OF TRANS. BLOCK]	71	47	1000111	107	G	119	77	1110111	167	w
24	18	11000	30	[CANCEL]	72	48	1001000	110	H	120	78	1111000	170	x
25	19	11001	31	[END OF MEDIUM]	73	49	1001001	111	I	121	79	1111001	171	y
26	1A	11010	32	[SUBSTITUTE]	74	4A	1001010	112	J	122	7A	1111010	172	z
27	1B	11011	33	[ESCAPE]	75	4B	1001011	113	K	123	7B	1111011	173	{
28	1C	11100	34	[FILE SEPARATOR]	76	4C	1001100	114	L	124	7C	1111100	174	
29	1D	11101	35	[GROUP SEPARATOR]	77	4D	1001101	115	M	125	7D	1111101	175	}
30	1E	11110	36	[RECORD SEPARATOR]	78	4E	1001110	116	N	126	7E	1111110	176	~
31	1F	11111	37	[UNIT SEPARATOR]	79	4F	1001111	117	O	127	7F	1111111	177	[DEL]
32	20	100000	40	[SPACE]	80	50	1010000	120	P					
33	21	100001	41	!	81	51	1010001	121	Q					
34	22	100010	42	"	82	52	1010010	122	R					
35	23	100011	43	#	83	53	1010011	123	S					
36	24	100100	44	\$	84	54	1010100	124	T					
37	25	100101	45	%	85	55	1010101	125	U					
38	26	100110	46	&	86	56	1010110	126	V					
39	27	100111	47	'	87	57	1010111	127	W					
40	28	101000	50	(88	58	1011000	130	X					
41	29	101001	51)	89	59	1011001	131	Y					
42	2A	101010	52	*	90	5A	1011010	132	Z					
43	2B	101011	53	+	91	5B	1011011	133	[
44	2C	101100	54	,	92	5C	1011100	134	\					
45	2D	101101	55	-	93	5D	1011101	135]					
46	2E	101110	56	.	94	5E	1011110	136	^					
47	2F	101111	57	/	95	5F	1011111	137	_					

Chaque code binaire (base 2) a une correspondance en décimale (base 10) que nous avons davantage l'habitude d'utiliser. Mais aussi en base hexadécimale (base 16) qui est très utilisée aussi en informatique.

Donner le code binaire et décimal généré lorsqu'on appuie sur les touches suivantes :

TOUCHE	Binaire	Décimal
B		
X		
a		
<		
f		
(
Barre espace		
7		
9		
=		

De même, un écran d'ordinateur est constitué de pixels de couleurs :



Chaque pixel est en réalité composé de 3 « sous » pixels : Rouge, Vert et Bleu.
En jouant avec l'intensité de chaque sous-pixel il est possible de reproduire une grande palette de couleur, c'est ce qu'on appelle le système **RGB** (Red, Green, Blue ou RVB en français).

Exemple pour quelques couleurs :

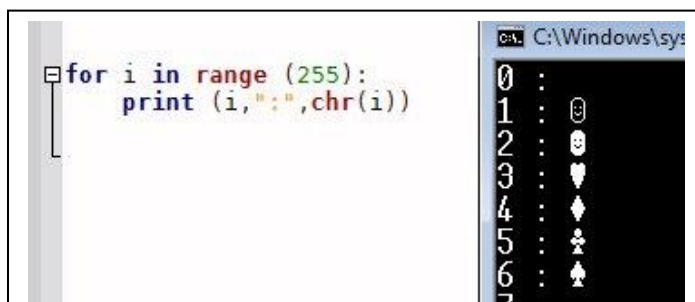
	Noir	Bleu	Jaune	Vert	Rouge	Blanc	Violet
R	0	120 0111 1000	255 1111 1111	120	255	255	203
G	0	120 0111 1000	255 1111 1111	255	120	225	143
B	0	255 1111 1111	0 0000 0000	120	120	255	252

Sachant qu'il y a 255 niveaux d'intensité pour chaque couleurs de base, on a au total $255 \times 255 \times 255 = 16\,581\,375$ couleurs possibles pour un pixel.

L'utilisation de la base 16 ou hexadécimale est très répandue en informatique. Voici par exemple un code source html-css qui permet de fixer la couleur d'une zone de texte :

```
#header h1 {
  margin-bottom: 0px;
  font-size: 28px;
  font-weight: bold;
  color: #A400FF      /* Violet */
}
```

Le nombre A400FF est au format hexadécimal et correspond à la couleur violette.
A4 : Niveau de rouge 00 : Niveau de vert FF: Niveau du bleu .

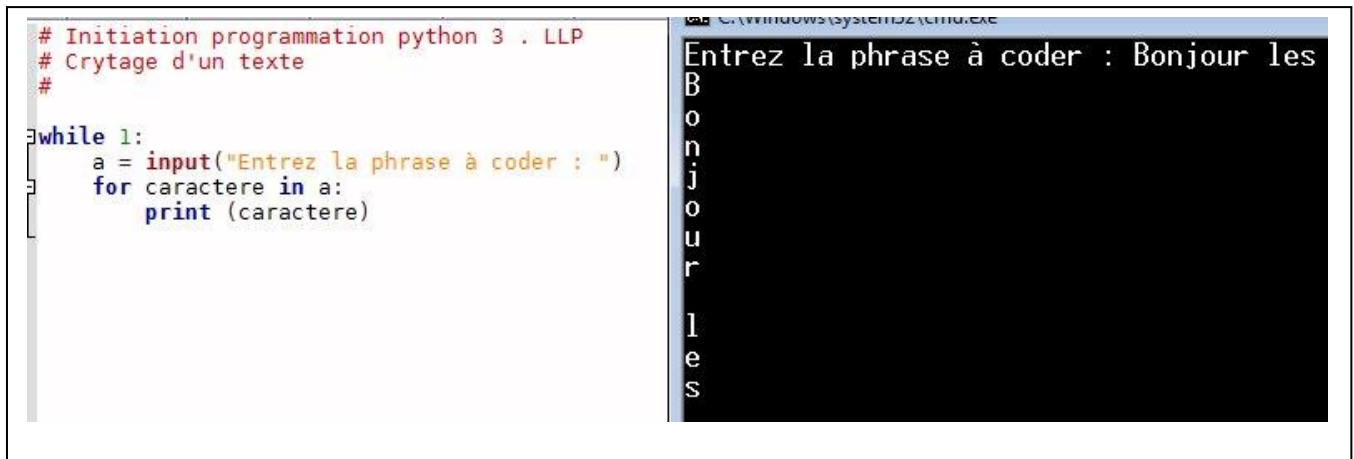


L'instruction **chr(valeur)** permet de récupérer le caractère à partir du code ASCII .
A l'inverse , l'instruction **ord("X")** permet de récupérer le code ascii à partir du caractère X.

II- Applications : Manipulation chaînes de caractères, encodage / cryptage.

Problématique : on souhaite crypter un texte afin d'en cacher le contenu et bien sur pouvoir ensuite le décrypter.

Ecrivons le programme suivant :

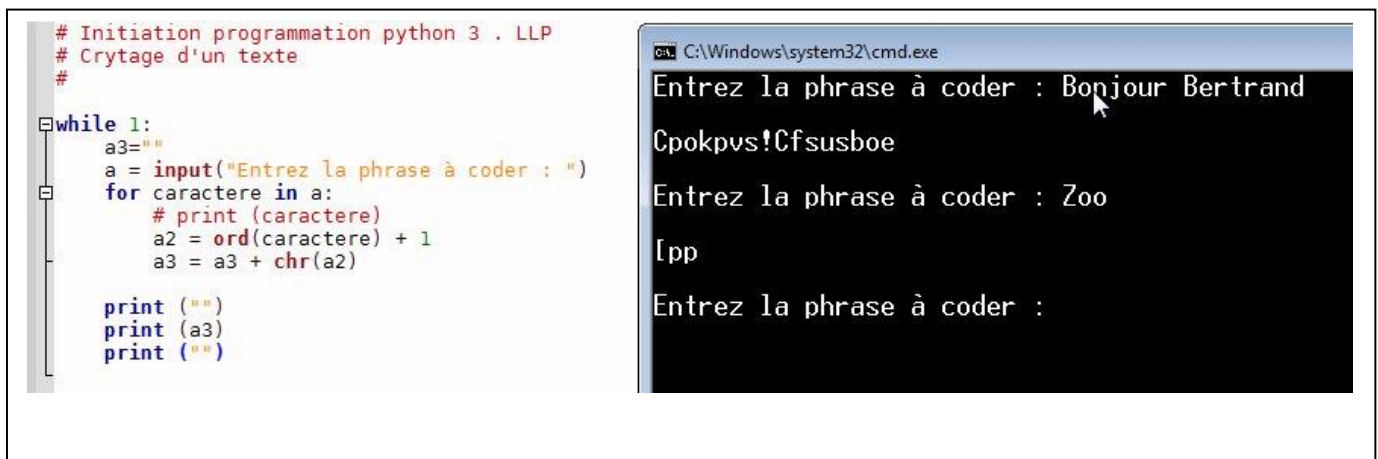


The image shows a Python script on the left and its execution in a terminal window on the right. The script is a simple loop that asks for a phrase and prints each character on a new line. The terminal shows the prompt 'Entrez la phrase à coder : Bonjour les' followed by the characters 'B', 'o', 'n', 'j', 'o', 'u', 'r', 'l', 'e', 's' printed vertically.

```
# Initiation programmation python 3 . LLP
# Cryptage d'un texte
#
while 1:
    a = input("Entrez la phrase à coder : ")
    for caractere in a:
        print (caractere)
```

While 1 : réalise une boucle toujours vraie (infinie)

for caractere in a : la variable caractere va prendre successivement la valeur de chaque caractère du texte, y compris les espaces



The image shows a Python script on the left and its execution in a terminal window on the right. The script is a loop that asks for a phrase, shifts each character by one in the ASCII table, and prints the result. The terminal shows the prompt 'Entrez la phrase à coder : Bonjour Bertrand', the output 'Cpokpvs!Cfsusboe', the prompt 'Entrez la phrase à coder : Zoo', the output '[pp', and the prompt 'Entrez la phrase à coder : '.

```
# Initiation programmation python 3 . LLP
# Cryptage d'un texte
#
while 1:
    a3=""
    a = input("Entrez la phrase à coder : ")
    for caractere in a:
        # print (caractere)
        a2 = ord(caractere) + 1
        a3 = a3 + chr(a2)

    print ("" )
    print (a3)
    print ("" )
```

1- Expliquer, à partir des données d'entrées (phrase à coder) et de sortie (phrase cryptée) et du tableau des codes ASCII, ce que fait le programme .

2- Proposer un programme qui va permettre de décoder le texte.

En entrée on aura le texte crypté (variable a3)

En sortie on affichera le texte décrypté

III- Programmation

Ci-dessous du code qui pourra vous servir plus tard en mini projet.

Il permet de faire de la saisie d'une touche du clavier au vol sans avoir à appuyer sur « entrée » ce qui est très pratique pour de petits jeux sympas ...

Modifier le pour ajouter des actions en fonction de la touche appuyée.

```
# CLAVIER NON BLOQUANT
# la base pour vos futurs jeux animés...
# python 3 LLP HB

from msvcrt import getch, kbhit
import sys,time

while True:
    if kbhit():          # Test si on appuyé sur une touche du clavier

        z = getch()      # lecture de la touche

        code = ord(z)     # récupère le code ascii de la touche
        print (code)

        # touche echap sortir de la boucle
        if code == 27:    # code ascii
            break
        if code == 97:
            print ("vous avez appuyé sur la touche a")
        if code == 98:
            print ("vous avez appuyé sur la touche b")

        if code == 224:   # on a appuyé sur une touche de direction
            z = getch()   # faut refaire une lecture pour connaitre laquelle
            code = ord(z) # récupère le code ascii de la touche de direction
            print (code)
            if code==77:
                print ("vous avez appuyé sur la touche fleche droite")
            if code==75:
                print ("vous avez appuyé sur la touche fleche gauche")

        # if z.decode()=="a":

    print ("hello")      # est exécuté toutes les 0,5s
    time.sleep(0.5)      # temporiser 0,5 seconde
```