

Rappels de cours

Les chaînes de caractères sont de type `str`. Bien qu'elles soient considérées comme un type de base, elles possèdent des fonctions et des **méthodes** propres aux types de données plus élaborés comme les **listes**. On peut en outre les parcourir avec une boucle `for`. On peut définir une chaîne de caractère avec des quotes ou des apostrophes. Tous les éléments de syntaxe ci-dessous sont valables pour les listes et les tuples.

Concaténation On peut **concaténer**, c'est à dire mettre bout à bout des chaînes de caractères :

```
>>>ch = 'Alan' + ' ' + 'Turing'
>>>ch
Alan Turing
```

Égalité En plus de la concaténation, d'autres opérateurs sont disponibles comme l'opérateur `!=` ou l'opérateur `==` qui permet de tester l'égalité de deux chaînes de caractères :

```
>>>'Atari' == 'Atari'
True
>>>'Amiga' == 'amiga'
False
```

Longueur La fonction `len` renvoie la longueur de la chaîne de caractère.

```
>>>len(ch)
11
```

Accès aux éléments On accède à ses éléments à l'aide des crochets et de l'index du caractère:

```
>>>ch[2]
'a'
```

Parcours Deux types de parcours, sur les éléments ou sur les indices:

```
>>>for elem in 'Atari':
    ...print(elem, end=" ")
'A t a r i'

>>>chaine = 'Linux'
>>>for i in range(len(chaine)):
    ...print(chaine[i], end=" ")
'L i n u x'
```

Les éléments de la liste sont affichés un par un, à chaque tour de boucle. Voir Introduction pour Python 3, §1.7.2 et §1.7.4

Notez que l'argument `end` permet juste d'éviter le retour à la ligne.

Exercices

1. Ecrire une fonction `nb_car` prenant deux paramètres `chaine` et `car` qui renvoie le nombre de fois où le caractère `car` apparaît dans la chaîne de caractère `chaine`.

```
>>>nb_car('atari', 'a')
2
```

2. Ecrire une fonction `etoiler` qui renvoie la chaîne de caractères prise en paramètre en insérant une étoile entre chaque élément.

```
>>>etoiler('Commodore')
'C*o*m*m*o*d*o*r*e'
```

3. Ecrire une fonction `trancher` prenant en paramètre `chaine`, `debut`, `fin` renvoyant les éléments situés entre les indices `debut` et `fin` de la chaîne de caractère `chaine`.

Exemple

```
>>>tranche('Amiga', 1, 3)
'mig'
```

	0	1	2	3	4
Seq	A	m	i	g	a

4. Ecrire une fonction `inverser` qui renvoie la chaîne de caractères prise en paramètre écrite à l'envers.

```
>>>inverse('Atari')
'iratA'
```

5. Ecrire une fonction `est_palindrome` qui détermine si la chaîne est un palindrome.

```
>>>est_palindrome('Amiga')
False
>>>est_palindrome('ADA')
True
```

6. Ecrire une fonction `dec2bin` qui prend en paramètre `entier` et renvoie sous forme de chaîne de caractère la représentation en base 2 de `entier`.

```
>>>dec2bin(12)
'1100'
```

7. Pour deux mots de même longueur, la distance de Hamming est le nombre de fois où les lettres sont différentes. Ecrire une fonction `distance_hamming` qui calcule la distance de Hamming entre deux mots de même longueur. Utilisez une assertion pour vérifier que ces deux mots sont de même longueur.

Exemple

```
>>> distance_hamming('sapin', 'malin')
2
```

Exercice 2 Une molécule d'ADN est formé d'environ six milliards de nucléotides. Ces nucléotides sont au nombre de quatre et sont notés A, C, T, G. Une séquence d'ADN est donc un mot de la forme *AATGTAGGTA...*

Écrire une fonction `est_present` qui prend en paramètre une séquence d'ADN et un code formé de plusieurs nucléotides. Elle renvoie la position de la première occurrence ou -1 si le code n'est pas présent dans la séquence :

1. Pour un code à 3 nucléotides.
2. Pour n'importe quel code de longueur inférieure ou égale à celle de la séquence.

```
>>> est_present('CTTTGGATTGCA', 'TTG')
2
>>> est_present('CTTTGGATTGCA', 'TTA')
-1
```

Exercice 3 Chaque caractère est stocké sur l'ordinateur par un nombre entier, c'est le code ASCII/Unicode qui assure la correspondance entre cet entier et le caractère associé.

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Figure 1: Code ASCII

1. Se documenter sur les fonctions Python `ord` et `chr`.
2. Écrire une fonction `majuscules` prenant en paramètre une chaîne de caractère `ch` ne contenant que des lettres de l'alphabet et qui renvoie cette chaîne en majuscule.

```
>>> majuscules('Linux')
'LINUX'
```

Cours : Méthodes Les méthodes sont des fonctions que l'on utilise avec la *notation pointée* sur l'objet auquel elles s'appliquent. Les chaînes de caractères en Python disposent de nombreuses méthodes dont les méthodes `lower` et `upper` qui permettent de renvoyer respectivement la chaîne en minuscules et en majuscules.

```
>>> 'LinuX'.lower()
'linux'
>>> ch = 'amiga'
```

```
>>> ch.upper()  
'AMIGA'  
>>> ch  
'amiga'
```

Cet exercice se traite bien sûr sans utiliser ces méthodes.



Figure 2: *Le logo de Commodore adapté à son fameux C64*

Pionnier dès 1977 avec le Personal Electronic Transactor, concurrent de l'Apple II et premier ordinateur tout en un : clavier, écran, lecteur de K7, la marque Commodore créée en 1955 a fabriqué des ordinateurs familiaux jusqu'au milieu des années 90 dont les fameux Commodore 64 et Amiga 500. Ces machines étaient livrées avec le système d'exploitation de la marque, notamment AmigaOS.

Exercice 4 Ecrire une fonction `starwars` prenant en paramètre `nom`, `prenom`, `nom2`, `ville` qui renvoie sous forme de chaîne de caractère le pseudonyme Star Wars de la personne. Ce pseudonyme est formé d'un prénom et d'un nom. Suivez la consigne :

Prénom : Les trois premières lettres de votre nom suivies des deux premières lettres de votre prénom.

Nom : Les deux premières lettres du nom de jeune fille de votre mère suivies des trois premières lettres de votre ville de naissance.

```
>>> starwars(Sanchez, Mila, Maillet, Bouvigny )  
'SanMi MaBou'
```