

# Cours de Python – 1 : L'ordinateur

## 1) Un ordinateur est une machine / un ordinateur est un outil

Un **ordinateur** est un **système de traitement de l'information programmable** tel que défini par Turing et qui fonctionne par la **lecture séquentielle d'un ensemble d'instructions**, organisées en **programmes**, qui lui font exécuter des **opérations logiques et arithmétiques**.

Sa structure physique actuelle fait que toutes les opérations reposent sur la **logique binaire** et sur des nombres formés à partir de **chiffres binaires**.

Dès sa mise sous tension, un ordinateur exécute, l'une après l'autre, des instructions qui lui font lire, manipuler, puis réécrire un ensemble de données déterminées par une **mémoire morte d'amorçage** (Boot ROM, BIOS, UEFI, etc.).

Les **données** à manipuler sont obtenues, soit par la **lecture de mémoires**, soit par la **lecture d'information en provenance de périphériques internes ou externes** (déplacement d'une souris, touche appuyée sur un clavier, température, vitesse, compression...).

Une fois utilisés, ou manipulés, les **résultats** sont écrits, soit dans des **mémoires**, soit dans des **composants qui peuvent transformer une valeur binaire en une action physique** (écriture sur une imprimante ou sur un moniteur, accélération ou freinage d'un véhicule, changement de température d'un four...).

## 2) La carte mère

La **carte mère** est le cœur de tout ordinateur compatible PC. Elle est essentiellement composée de **circuits imprimés** et de **ports de connexion** qui assurent la **liaison de tous les composants et périphériques propres à un micro-ordinateur** (disques durs, mémoire vive, microprocesseur, cartes filles, etc.) afin qu'ils puissent être **reconnus et configurés par le microprocesseur grâce au programme contenu dans le BIOS**, au travers de la carte, afin d'assurer la **configuration et le démarrage correct de tous les équipements**.



## 3) Le microprocesseur

Le processeur est connu comme étant le cerveau de l'ordinateur, il **exécute les instructions** et **traite les données des programmes**. Il est mesuré en Hertz. Lorsque ce dernier est élevé, le signal de l'exécution devient de plus en plus rapide.

Un **microprocesseur** est un processeur dont tous les composants ont été suffisamment **miniaturisés** pour être regroupés dans un unique boîtier.

La **complexité** de son architecture se mesure par le **nombre de transistors** qu'il contient.

La **puissance du microprocesseur** qui s'exprime en « **millions d'instructions par seconde** » (MIPS).

La **vitesse de l'horloge** : le rôle de l'horloge est de cadencer le rythme du travail du microprocesseur. Plus la vitesse de l'horloge augmente, plus le microprocesseur effectue d'instructions en une seconde.



ARM60



Intel Core 2 Duo

#### 4) Les mémoires d'un ordinateur :

##### La mémoire vive d'un ordinateur : les cartes mémoire

Lorsque l'ordinateur est sous tension, le disque dur, qui stocke durablement les données, se lance et vient charger dans cette **mémoire vive** le système d'exploitation (Windows, OS X, Linux), puis les différentes applications que vous ouvrez (par exemple, un navigateur internet, ou un nouveau document). Et lorsque vous travaillez sur votre document, tant qu'il n'est pas enregistré sur le disque dur, c'est sur cette mémoire vive que la sauvegarde est effectuée.

Les cartes mémoires se présentent sous 2 formats aux appellations singulières : DIMM pour les ordinateurs de bureau et SO-DIMM pour les portables.

##### **1 million de fois plus rapide qu'un disque dur**

**L'avantage de cette mémoire vive, par rapport au disque dur, c'est qu'elle est très rapide.** Elle réagit au quart de tour pour transmettre des données à traiter au processeur et les exploiter. Elle peut faire transiter des données à plus de 10 Go/s, alors qu'un disque plafonne autour de 200 Mo/s en pratique. De même, le temps d'accès aux données, autrement dit son temps de réaction, se compte en nanosecondes, alors que pour un disque dur on parle de millisecondes.

##### **8 Go pour ne pas avoir à sortir les rames**

En revanche, contrairement au disque dur dont la capacité de stockage est colossale et peut atteindre plusieurs Téraoctets (1000 Gigaoctets), la mémoire vive se limite à 2, 4, 8, voire 16 Gigaoctets. 4 Go est une quantité toutefois suffisante pour des travaux de bureautique, mais il faut retenir que plus il y a de capacité et moins cette mémoire vive est engorgée. Dès qu'elle l'est, l'ordinateur va générer de la fausse mémoire vive sur le disque dur. Pour le coup, c'est à partir de ce moment que l'on peut dire que « ça rame » ! C'est pourquoi, lorsque l'on est fan de jeux ou d'arts graphiques, 8 Go présentent le strict minimum et 16 Go offrent plus de fluidité pour des jeux 3D de plus en plus gourmands en mémoire.

- **Rapidité d'accès**, essentielle pour fournir rapidement les données au processeur ;
- **Volatilité**, qui entraîne une **perte de toutes les données en mémoire dès que l'ordinateur cesse d'être alimenté en électricité.**

##### Le disque dur d'un ordinateur : la mémoire morte

L'ordinateur doit être muni d'un disque dur pour pouvoir stocker vos données. A la différence de la mémoire vive, le disque dur peut conserver toutes vos données le plus longtemps possible. Toutefois, s'il se détériore, les données seront perdues et vous perdrez tout ce que vous avez enregistré dans le disque dur. Il est préférable dans de cas d'avoir un stockage amovible pour conserver la copie de vos données.

## 5) Le BIOS

**Lors du démarrage, la carte mère a besoin de savoir quels périphériques lui sont connectés.**

Pour effectuer cette tâche, elle dispose d'un *firmware* initialement appelé BIOS (de l'anglais *Basic Input Output System* « signifiant système d'entrée/sortie de base ») ou sur les ordinateurs plus récent de son équivalent l'UEFI. L'un et l'autre sont contenus dans une **puce de « mémoire morte » soudée à même la carte mère**. Le microprocesseur lance ce code automatiquement lorsque la carte est mise sous tension — autrement dit, lorsque l'utilisateur allume son ordinateur.

### Le BIOS

Le microprocesseur utilise le code contenu dans le BIOS pour configurer chaque périphérique connecté à la carte mère (mémoire vive, disques durs, cartes d'extension, etc.). Si le microprocesseur ne peut pas afficher un message d'erreur, ni lancer le « configurateur du BIOS », la carte émet une série de *bips* qui permettent d'informer l'utilisateur du problème empêchant de continuer (entre autres si l'écran, le clavier ou la souris n'ont pas été détectés) — ces signaux sont propre à chaque carte. La signification de l'erreur liée à cette série de *bips* est précisée dans le manuel de la carte mère.

Lorsque tous les périphériques ont été configurés, le microprocesseur exécute les instructions contenues dans le *Master Boot Record* trouvé sur le périphérique identifié comme celui contenant le premier système disponible contenu dans un des périphériques de stockage de masse (CDROM, disque dur, SSD, clé USB, etc.).

### L'UEFI

L'UEFI est le nouveau type de BIOS lié aux évolutions technologiques depuis les années 2000. Un consortium de constructeurs a mis au point un nouveau standard de *firmware* (micrologiciel intermédiaire entre les éléments composants de la carte mère et le système d'exploitation). L'UEFI (de l'anglais *Unified Extensible Firmware Interface*) offre quelques avantages sur le BIOS : fonctionnalités réseau en standard, interface graphique haute résolution, gestion intégrée d'installations multiples de systèmes d'exploitation et affranchissement de la limite des disques à 2,2 To. L'UEFI est écrit en langage C

## 6) Le code binaire, le bit, l'octet

Le système binaire est utile pour représenter le fonctionnement de l'électronique numérique utilisée dans les ordinateurs. Il est donc utilisé par les langages de programmation de bas niveau.

Le système binaire le plus courant est la **base deux** mathématique, permettant de représenter des nombres à l'aide de la numération de position avec seulement deux chiffres : le **0** et le **1**.

Dans ce type de codage, chaque nombre est représenté de façon unique par une **suite ordonnée de chiffres**.

$$\text{Ex : } (1)_{10} = 1 \times 2^0 = (1)_2$$

$$(2)_{10} = 1 \times 2^1 + 0 \times 2^0 = (10)_2$$

$$(3)_{10} = 1 \times 2^1 + 1 \times 2^0 = (11)_2$$

$$(4)_{10} = 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 = (100)_2$$

$$(5)_{10} = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = (101)_2$$

$$(6)_{10} = 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = (110)_2$$

A l'origine, il s'agissait de diodes qui s'allumaient ou s'éteignaient dans un style de **morse numérique**.

Dans la théorie de l'information, **un bit est la quantité minimale d'information transmise par un message**, et constitue à ce titre l'**unité de mesure de base de l'information** en informatique

**Un bit est l'élément de base** avec lequel travaille l'ordinateur : **sa valeur est 1 ou 0.**

→ on peut comparer ça à : on/off ; vrai/faux ; marche/arrêt ; ...

Valeur décimale	Valeur binaire
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111
...	
255	11111111

Décimal	Binaire			
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0

$$8 \text{ bits} = 2^8 = 256$$

$$16 \text{ bits} = 2^{16} = 65\,536$$

Je me suis arrêté sur la valeur 255 qui correspond à 8 bits (vous voyez en face du 255 il y a 8 chiffres 1).

Ces 8 bits correspondent à ce que l'on appelle **un octet** (venant de octal : 8). Pour votre culture ...

$$\text{Ainsi } (101011)_2 = 1 \times 2^7 + 0 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 128 + 32 + 8 + 2 + 1 = (171)_{10}$$

$$(234)_{10} = 2 \times 10^2 + 3 \times 10^1 + 4 \times 10^0$$

**Un octet** (ou byte) est un multiple de **8 bits** codant une information : ex : (11101011).

En fait l'ordinateur ne calcule jamais sur 1 bit à la fois, mais sur un ou plusieurs octets (ce qui fait 8 bits ou plus, mais toujours des multiples de 8 bits).

**Les premiers ordinateurs personnels étaient 8 bits** (ils ne comptaient que sur 1 octet à la fois)

→ un octet permet de représenter  $2^8$  nombres, soit 256 valeurs différentes.

→ un octet permet de coder des valeurs numériques ou jusqu'à 256 caractères différents.

Les systèmes d'exploitation ont évolué pour passer de 8 bits à 16 bits (la série des windows 1 à Windows 3.xx), à 32 bits (Windows 95 à Windows 7 32 bits) et enfin à 64 bits (Windows XP 64 bits à Windows 8).

Les nouveaux ordinateurs sont 64 bits (c'est à dire qu'ils calculent sur 8 octets à la fois (8 octets x 8 bits = 64 bits)). Les éléments qui composent l'ordinateur (Processeur, bus de données, etc) sont en 64 bits. Ils peuvent donc ainsi adresser plus d'informations en une seule fois (2 fois plus qu'en 32 bits).

L'octet est couramment utilisé pour indiquer la **capacité de mémorisation des mémoires**.

$$\text{Le plus grand entier naturel que l'on puisse coder sur 8 bits est : } \left( \begin{matrix} 11\dots1 \\ 8 \text{ un} \end{matrix} \right)_2 = 255$$

Le plus grand entier naturel que l'on puisse coder sur 16 bits est :  $\left( \begin{matrix} 11\dots 1 \\ 16 \text{ un} \end{matrix} \right)_2 = 65\,536$

Le plus grand entier naturel que l'on puisse coder sur 64 bits est :  $\left( \begin{matrix} 11\dots 1 \\ 64 \text{ un} \end{matrix} \right)_2 = 2^{64} - 1$

→ Démonstrations :  $\left( \begin{matrix} 11\dots 1 \\ 8 \text{ un} \end{matrix} \right)_2 + 1 = \left( \begin{matrix} 10\dots 0 \\ 8 \text{ un} \end{matrix} \right)_2 = 1 \times 2^8 = 256$  donc  $\left( \begin{matrix} 11\dots 1 \\ 8 \text{ un} \end{matrix} \right)_2 = 255$

$\left( \begin{matrix} 11\dots 1 \\ 64 \text{ un} \end{matrix} \right)_2 + 1 = \left( \begin{matrix} 10\dots 0 \\ 64 \text{ un} \end{matrix} \right)_2 = 2^{64}$  donc  $\left( \begin{matrix} 11\dots 1 \\ 64 \text{ un} \end{matrix} \right)_2 = 2^{64} - 1$

→ Ordre de grandeur du résultat :

$$2^{10} = 1024 \approx 10^3 \text{ donc } 2^{64} = (2^{10})^6 \times 2^4 \approx (10^3)^6 \times 16 \approx 1,6 \times 10^{19}$$

$$2^{64} = 18446744073709551614 \approx 1,84 \times 10^{19}$$

## 7) Le système hexadécimal

Le code binaire génère des messages souvent beaucoup trop longs à exploiter, ce qui sature les mémoires et ralentit le fonctionnement de la machine.

On utilise le langage hexadécimal pour alléger les écritures, en base 16 alphanumérique, avec 16 symboles : les dix premiers chiffres et les lettres A à F pour les six suivants.

L'**hexadécimal** est une écriture compacte du binaire.

L'**hexadécimal** est très utilisé en informatique car il facilite les conversions en binaire.

Un nombre écrit en **hexadécimal** permet de représenter 4 bits, autrement dit un nibble.

Base décimale	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Base hexadécimale	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Base binaire	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

**Convertir de la base 2 à la base 16** : on fait des groupements de 4 :

$$0001\,1011 = 1B$$

$$1001\,1100\,0111\,0101 = 9C75$$

$$(0010\,1010\,1101\,0101)_2 = (2AD5)_{16}$$

**Convertir de la base 16 à la base 2** :

$$A95D7 = 1010\,1001\,0101\,1101\,0111$$

**Convertir de la base 10 en système hexadécimal** :

$$(27)_{10} = (11011)_2$$

$$\rightarrow \text{en base 16 : } (27)_{10} = 1 \times 16^1 + 11 \times 16^0 = (1B)_{16}$$

$$(62)_{10} = 3 \times 16^1 + 14 \times 16^0 = (3E)_{16}$$

**Convertir du système hexadécimal en base 10** :

$$(3D8)_{16} = 3 \times 16^2 + 13 \times 16^1 + 8 \times 16^0 = 768 + 208 + 8 = (984)_{10}$$

$$(FB3)_{16} = F \times 16^2 + B \times 16^1 + 3 \times 16^0 = 15 \times 256 + 11 \times 16 + 3 \times 1 = (4019)_{10}$$

$$4D5_{16} = 4 \times 16^2 + D \times 16^1 + 5 \times 16^0 = 4 \times 256 + 13 \times 16 + 5 \times 1 = 1\ 237_{10}$$

$$15AACF7_{16} = 1 \times 16^6 + 5 \times 16^5 + A \times 16^4 + A \times 16^3 + C \times 16^2 + F \times 16^1 + 7 \times 16^0 = 22\ 719\ 735_{10}$$

## 8) Le code ASCII

La mémoire de l'ordinateur conserve toutes les données sous forme numérique. Il n'existe pas de méthode pour stocker directement les caractères. **Chaque caractère possède donc son équivalent en code numérique** : c'est le **code ASCII** (*American Standard Code for Information Interchange* - traduisez « Code Américain Standard pour l'Echange d'Informations »). Le code ASCII de base représentait les caractères sur 7 bits (c'est-à-dire 128 caractères possibles, de 0 à 127).

- Les codes 0 à 31 ne sont pas des caractères. On les appelle *caractères de contrôle* car ils permettent de faire des actions telles que :
  - retour à la ligne (CR)
  - Bip sonore (BEL)
- Les codes 65 à 90 représentent les majuscules
- Les codes 97 à 122 représentent les minuscules

## The ASCII code

American Standard Code for Information Interchange

ASCII control characters			ASCII printable characters									Extended ASCII characters											
DEC	HEX	Simbolo ASCII	DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo			
00	00h	NULL (carácter nulo)	32	20h	espacio	64	40h	@	96	60h	`	128	80h	Ç	160	A0h	á	192	C0h	Ł	224	E0h	Ó
01	01h	SOH (inicio encabezado)	33	21h	!	65	41h	A	97	61h	a	129	81h	ù	161	A1h	í	193	C1h	ł	225	E1h	ô
02	02h	STX (inicio texto)	34	22h	"	66	42h	B	98	62h	b	130	82h	é	162	A2h	ô	194	C2h	ŧ	226	E2h	ó
03	03h	ETX (fin de texto)	35	23h	#	67	43h	C	99	63h	c	131	83h	à	163	A3h	ú	195	C3h	Ŧ	227	E3h	ô
04	04h	EOT (fin transmisión)	36	24h	\$	68	44h	D	100	64h	d	132	84h	â	164	A4h	ñ	196	C4h	Ũ	228	E4h	õ
05	05h	ENQ (enquiry)	37	25h	%	69	45h	E	101	65h	e	133	85h	ã	165	A5h	Ñ	197	C5h	Ű	229	E5h	ö
06	06h	ACK (acknowledgement)	38	26h	&	70	46h	F	102	66h	f	134	86h	ä	166	A6h	°	198	C6h	Ź	230	E6h	ű
07	07h	BEL (timbre)	39	27h	'	71	47h	G	103	67h	g	135	87h	ç	167	A7h	º	199	C7h	Ż	231	E7h	Ű
08	08h	BS (retroceso)	40	28h	(	72	48h	H	104	68h	h	136	88h	è	168	A8h	¿	200	C8h	Ł	232	E8h	Ų
09	09h	HT (tab horizontal)	41	29h	)	73	49h	I	105	69h	i	137	89h	ê	169	A9h	¸	201	C9h	ł	233	E9h	Ŵ
10	0Ah	LF (salto de línea)	42	2Ah	*	74	4Ah	J	106	6Ah	j	138	8Ah	ë	170	AAh	ˆ	202	CAh	Ł	234	EAh	Ŷ
11	0Bh	VT (tab vertical)	43	2Bh	+	75	4Bh	K	107	6Bh	k	139	8Bh	ì	171	ABh	˜	203	CBh	ł	235	EBh	Ÿ
12	0Ch	FF (form feed)	44	2Ch	,	76	4Ch	L	108	6Ch	l	140	8Ch	í	172	ACH	¼	204	Ch	Ł	236	ECh	Ź
13	0Dh	CR (retorno de carro)	45	2Dh	-	77	4Dh	M	109	6Dh	m	141	8Dh	î	173	ADh	½	205	CDh	ł	237	EDh	Ż
14	0Eh	SO (shift Out)	46	2Eh	.	78	4Eh	N	110	6Eh	n	142	8Eh	Ë	174	Aeh	¾	206	CEh	ł	238	Eeh	Ź
15	0Fh	SI (shift In)	47	2Fh	/	79	4Fh	O	111	6Fh	o	143	8Fh	Ä	175	Afh	»	207	Cfh	ł	239	Efh	Ź
16	10h	DLE (data link escape)	48	30h	0	80	50h	P	112	70h	p	144	90h	Å	176	B0h	»	208	D0h	ł	240	F0h	Ź
17	11h	DC1 (device control 1)	49	31h	1	81	51h	Q	113	71h	q	145	91h	æ	177	B1h	»	209	D1h	ł	241	F1h	Ź
18	12h	DC2 (device control 2)	50	32h	2	82	52h	R	114	72h	r	146	92h	Æ	178	B2h	»	210	D2h	ł	242	F2h	Ź
19	13h	DC3 (device control 3)	51	33h	3	83	53h	S	115	73h	s	147	93h	ø	179	B3h	»	211	D3h	ł	243	F3h	Ź
20	14h	DC4 (device control 4)	52	34h	4	84	54h	T	116	74h	t	148	94h	ö	180	B4h	»	212	D4h	ł	244	F4h	Ź
21	15h	NAK (negative acknowle.)	53	35h	5	85	55h	U	117	75h	u	149	95h	ò	181	B5h	»	213	D5h	ł	245	F5h	Ź
22	16h	SYN (synchronous idle)	54	36h	6	86	56h	V	118	76h	v	150	96h	ú	182	B6h	»	214	D6h	ł	246	F6h	Ź
23	17h	ETB (end of trans. block)	55	37h	7	87	57h	W	119	77h	w	151	97h	û	183	B7h	»	215	D7h	ł	247	F7h	Ź
24	18h	CAN (cancel)	56	38h	8	88	58h	X	120	78h	x	152	98h	ÿ	184	B8h	»	216	D8h	ł	248	F8h	Ź
25	19h	EM (end of medium)	57	39h	9	89	59h	Y	121	79h	y	153	99h	ÿ	185	B9h	»	217	D9h	ł	249	F9h	Ź
26	1Ah	SUB (substitute)	58	3Ah	:	90	5Ah	Z	122	7Ah	z	154	9Ah	ÿ	186	BAh	»	218	DAh	ł	250	FAh	Ź
27	1Bh	ESC (escape)	59	3Bh	;	91	5Bh	[	123	7Bh	{	155	9Bh	ÿ	187	BBh	»	219	DBh	ł	251	FBh	Ź
28	1Ch	FS (file separator)	60	3Ch	<	92	5Ch	\	124	7Ch		156	9Ch	ÿ	188	BCh	»	220	DCh	ł	252	FCh	Ź
29	1Dh	GS (group separator)	61	3Dh	=	93	5Dh	]	125	7Dh	}	157	9Dh	ÿ	189	BDh	»	221	DDh	ł	253	FDh	Ź
30	1Eh	RS (record separator)	62	3Eh	>	94	5Eh	^	126	7Eh	~	158	9Eh	x	190	BEh	»	222	DEh	ł	254	FEh	Ź
31	1Fh	US (unit separator)	63	3Fh	?	95	5Fh	-				159	9Fh	f	191	BFh	»	223	DFh	ł	255	FFh	Ź

Pour le dire autrement, l'ordinateur gère tous les caractères comme des nombres en base 16.