

# Introduction à l'informatique

## 1 - Introduction

*Julien Deantoni*

# Contenue de l'UE (non contractuel)

- 1) Introduction, notion de variables, de type de données simple et structuré
- 2) Notion d'algorithme, de boucles et de branchements conditionnels
- 3) Logique Booléenne et comparaison, lecture/écriture dans un fichier
- 4) Écrire des fonctions afin de rendre le code plus lisible et réutilisable
- 5) Contrôle continu intermédiaire
- 6) Introduction au web, structure d'une page en web en HTML
- 7) Modifier le style d'une structure HTML en CSS
- 8) Résumé
- 9) Contrôle continu final

# Intervenants

- Sarah Antier (Sarah.ANTIER@univ-cotedazur.fr) ,
- Julien Deantoni (Julien.DEANTONI@univ-cotedazur.fr) ,
- Franck Guingne (Franck.GUINGNE@univ-cotedazur.fr) ,
- Stéphane Jeannin (Stephane.JEANNIN@univ-cotedazur.fr)

	Lundi	Mardi	Mercredi	Jeudi	Vendredi
8:00-10:00		TD G1 Elec/PPPE M13 (→ 10/10) puis M37  Jeannin	TP G5 Oui-Si PV 316-317  Antier - Deantoni		
10:15-12:15	TD G2 MIASH2 M13  Jeannin		TP G2 MIASH2 PV 316-317  Antier - Deantoni	TD G4 LAS-IN M12  Jeannin	TD G3 MIASH 3 M35  Deantoni
13:15-15:15		TD G5 Oui-Si M12  Deantoni	TP G1 Elec/PPPE PV 316-317  Jeannin - Guingne		TP G4 LAS-IN PV 302  Deantoni
15:30-17:30		TP G3 MIASH3 PV 316-317  Deantoni - Jeannin	CM Amphi Physique  Deantoni		

# Plan du cours

- Introduction à l'informatique, pourquoi ?
- Informatique, ordinateur, processeur, système d'exploitation, application, application web ?
- Programmation et langage informatique?
- Structuration de l'information
  - Sur le disque dur
    - fichiers, répertoires, chemins
  - En mémoire vive
    - variables, types (simples et structurés)

# Introduction à l'informatique, pourquoi ?

- Vous permettre de toucher du doigt ce qu'est la science informatique
- Vous incitez à automatiser des tâches dans les sciences qui vous concernent, comme e.g.
  - Ajouter une ligne dans tous les fichiers d'un répertoire
  - Créer la page web représentant une galerie photo de toutes vos photos
  - Créer des analyses ou la visualisation automatiques de résultats
- Vous faire passer du statut de simple utilisateur de l'ordinateur au statut de développeurs de programmes informatiques
- Parce que la programmation est une discipline difficilement contournable aujourd'hui !

# Informatique, ordinateur, système d'exploitation, application, application web, ... ?

- Définitions

L'informatique est un domaine d'activité **scientifique**, **technique**, et **industriel** concernant le **traitement automatique de l'information numérique** par l'exécution de **programmes informatiques** hébergés par des dispositifs électriques-électroniques : des **systèmes embarqués**, des **ordinateurs**, des **robots**, des **automates**, etc.

Ces champs d'application peuvent être séparés en deux branches :

- **théorique** : concerne la définition de concepts et modèles ;
- **pratique** : s'intéresse aux **techniques** concrètes de mise en œuvre.



WIKIPÉDIA

## informatique

nom féminin

(de information et automatique)



1. Science du traitement automatique et rationnel de l'information considérée comme le support des connaissances et des communications.
2. Ensemble des applications de cette science, mettant en œuvre des matériels (ordinateurs) et des logiciels.

# Informatique, ordinateur, système d'exploitation, application, application web, ... ?

- Définitions

L'informatique est un domaine d'activité scientifique, technique, et industriel concernant le traitement automatique de l'information numérique par l'exécution de programmes informatiques hébergés par des dispositifs électriques-électroniques : des systèmes embarqués, des ordinateurs, des robots, des automates, etc.

Ces champs d'application peuvent être séparés en deux branches :

- théorique : concerne la définition de concepts et modèles ;
- pratique : s'intéresse aux techniques concrètes de mise en œuvre.



WIKIPÉDIA

## informatique

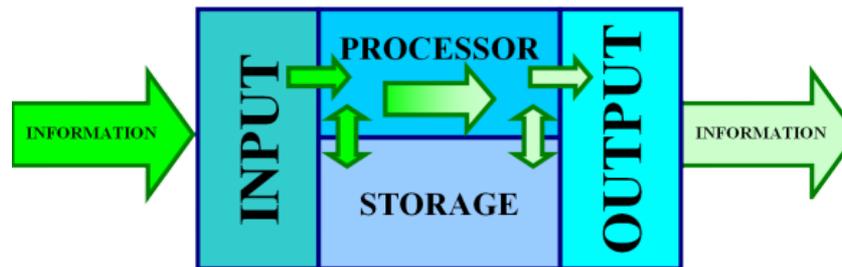
nom féminin

(de information et automatique)



1. Science du traitement automatique et rationnel de l'information considérée comme le support des connaissances et des communications.

2. Ensemble des applications de cette science, mettant en œuvre des matériels (ordinateurs) et des logiciels.



Système de traitement de l'information

# Informatique, ordinateur, système d'exploitation, application, application web, ... ?

- Definitions

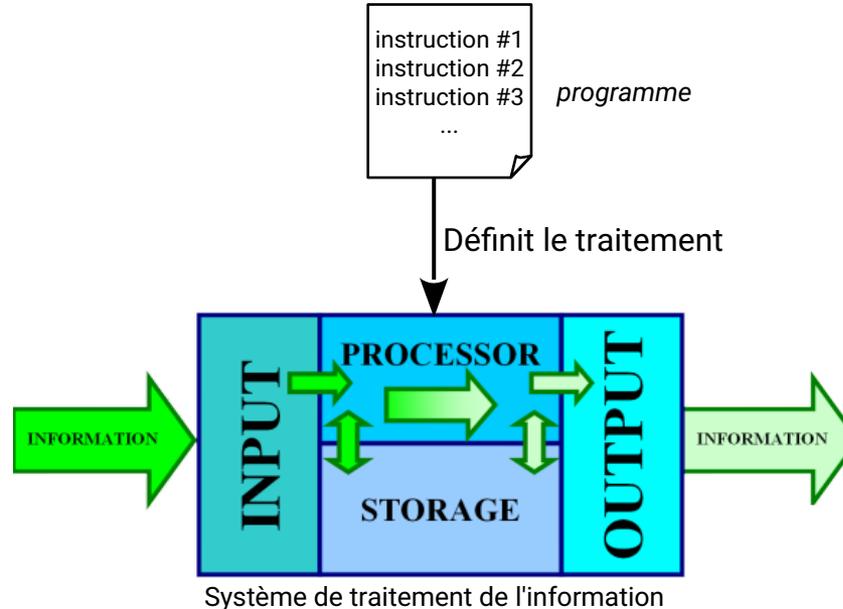
Un ordinateur est un **système de traitement de l'information programmable** tel que défini par Alan Turing et qui fonctionne par la lecture séquentielle d'un ensemble d'**instructions**, organisées en **programmes**, qui lui font exécuter des opérations **logiques** et **arithmétiques**. Sa structure physique actuelle fait que toutes les opérations reposent sur la **logique binaire** et sur des nombres formés à partir de **chiffres binaires**.



WIKIPÉDIA

G. Santini, J.-C. Dubacq  
<https://lipn.univ-paris13.fr/~dubacq/pdf/m1101/sys.pdf>

Machine électronique programmable capable de réaliser des calculs logiques sur des nombres binaires.



# Informatique, ordinateur, système d'exploitation, application, application web, ... ?

- Definitions

Un ordinateur est un **système de traitement de l'information programmable** tel que défini par Alan 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**.

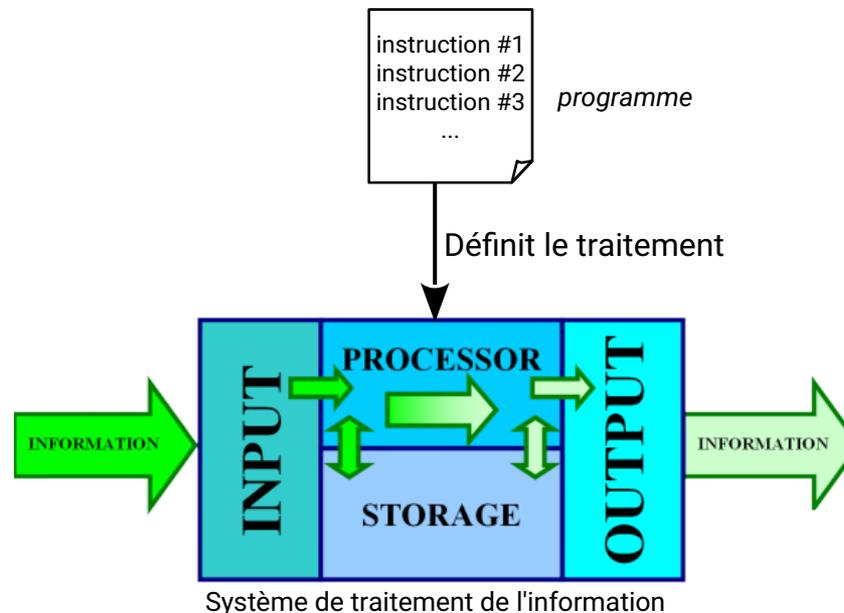


WIKIPÉDIA

Machine électronique programmable capable de réaliser des calculs logiques sur des nombres binaires.

G. Santini, J.-C. Dubacq

<https://lipn.univ-paris13.fr/~dubacq/pdf/m1101/sys.pdf>



# Informatique, ordinateur, système d'exploitation, application, application web, ... ?

- Definitions

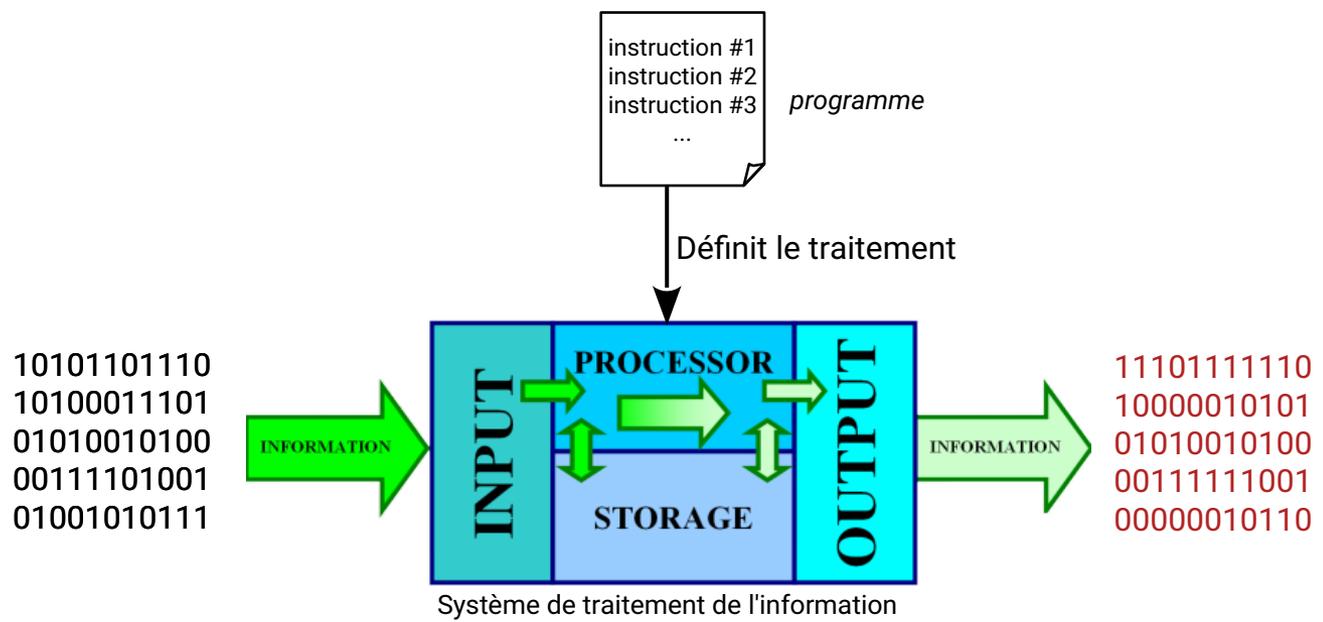
Un ordinateur est un système de traitement de l'information programmable tel que défini par Alan Turing et qui fonctionne par la lecture séquentielle d'un ensemble d'instructions, organisées en programmes, qui lui font exécuter des opérations logiques et arithmétiques. Sa structure physique actuelle fait que toutes les opérations reposent sur la logique binaire et sur des nombres formés à partir de chiffres binaires.



WIKIPÉDIA

Machine électronique programmable capable de réaliser des calculs logiques sur des nombres binaires.

G. Santini, J.-C. Dubacq  
<https://lipn.univ-paris13.fr/~dubacq/pdf/m1101/sys.pdf>



# Nombre binaire

- C'est un nombre écrit en **base 2**, c'est à dire avec uniquement 2 **symboles** : le symbole 0 et le symbole 1.
- Habituellement, l'humain écrit les nombres en **base 10**, c'est à dire avec 10 **symboles** que sont **0,1,2,3,4,5,6,7,8,9**
- En base **10** nous lisons naturellement les nombres, par exemple 1259. En fait ce que nous faisons, c'est que la **position** de chaque **symbole** permet de connaître sa valeur. Dans 1259, 9 est à la **position 0**, 5 à la **position 1**, etc. Le calcul est alors le suivant :
  - $\underset{3}{1} \underset{2}{2} \underset{1}{5} \underset{0}{9}_{10} = 9 \times 10^0 + 5 \times 10^1 + 2 \times 10^2 + 1 \times 10^3 = 9 + 50 + 200 + 1000$
- La même règle s'applique peu importe la **base**. En **base 2** nous avons
  - $\underset{7}{1} \underset{6}{0} \underset{5}{0} \underset{4}{1} \underset{3}{0} \underset{2}{0} \underset{1}{1} \underset{0}{1}_2 = 1 \times 2^0 + 1 \times 2^1 + 0 \times 2^2 + 0 \times 2^3 + 1 \times 2^4 + 0 \times 2^5 + 0 \times 2^6 + 1 \times 2^7$   
 $= 1 + 2 + 0 + 0 + 8 + 0 + 0 + 128$   
 $= 139_{10}$

# Nombre hexadécimal

- C'est un nombre écrit en **base 16**, c'est à dire avec 16 **symboles** : les symboles sont **F, E, D, C, B, A, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0**.
- La même règle s'applique peu importe la **base**. En **base 16** nous avons

$$\begin{aligned} \underset{\substack{1 \\ 0}}{A8}_{16} &= 8 \times 16^0 + 10 \times 16^1 \\ &= 8 + 160 \\ &= 168_{10} \end{aligned}$$

- 16 est une puissance de 2 :  $2^4$ , c'est donc simple de passer d'un nombre hexadécimal à un nombre binaire

$$= \begin{array}{c} A \quad 8 \\ \diagdown \quad \diagup \\ 1010 \quad 1000 \\ \diagup \quad \diagdown \\ 16 \end{array} \quad = 8 + 128 + 32 = 168_{10}$$

⇒ les nombres hexadécimaux sont souvent utilisés comme une représentation compacte des nombres binaires

# Octet (Bytes)

- Un Octet (Bytes en anglais) est un nombre binaire codé avec 8 bits, représentable avec 2 symboles en hexadécimal.
- Valeur maximal d'un octet ?
  - $1111\ 1111_2$  ou  $FF_{16}$   
 $\Rightarrow 255$  (i.e.,  $15 \cdot 16 + 15$  ou  $2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0$ )
- Remarque : En informatique on représente habituellement les nombres binaires et hexadécimaux en préfixant le nombre par un caractère représentant la base choisie sauf pour les nombres décimaux qui car la base 10 est celle par défaut:
  - $0b11111111 == 0xFF == 255$

# Informatique, ordinateur, système d'exploitation, application, application web, ... ?

- Définitions

Un ordinateur est un système de traitement de l'information programmable tel que défini par Alan 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.



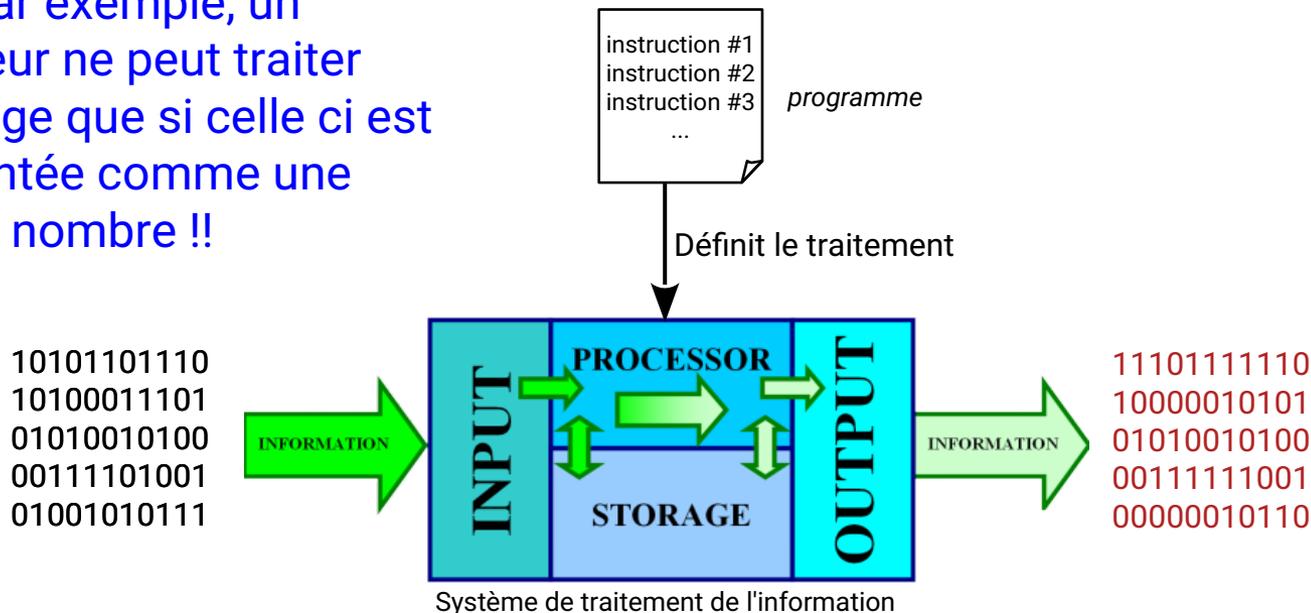
WIKIPÉDIA

Machine électronique programmable capable de réaliser des calculs logiques sur des nombres binaires.

G. Santini, J.-C. Dubacq

<https://lipn.univ-paris13.fr/~dubacq/pdf/m1101/sys.pdf>

Donc, par exemple, un ordinateur ne peut traiter une image que si celle-ci est représentée comme une suite de nombres !!



# Informatique, ordinateur, système d'exploitation, application, application web, ... ?

- Definitions

Un ordinateur est un **système de traitement de l'information programmable** tel que défini par Alan 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**.

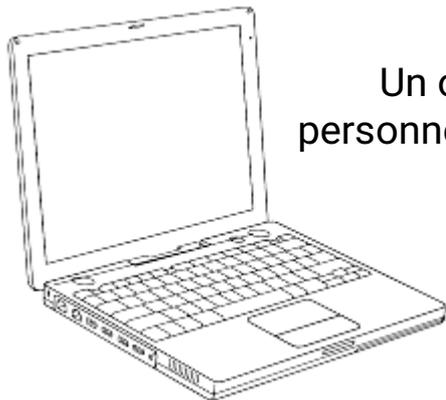


WIKIPÉDIA

Machine électronique programmable capable de réaliser des calculs logiques sur des nombres binaires.

G. Santini, J.-C. Dubacq

<https://lipn.univ-paris13.fr/~dubacq/pdf/m1101/sys.pdf>



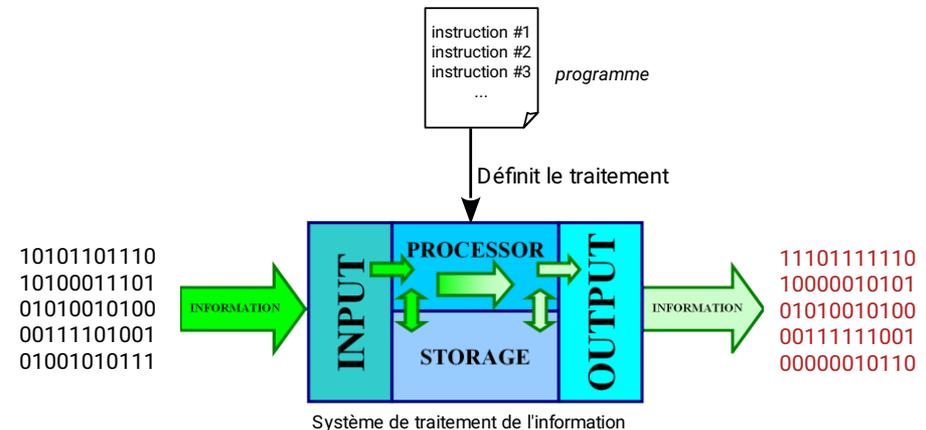
Un ordinateur personnel « classique »



concret

+ de nombreux *périphériques*

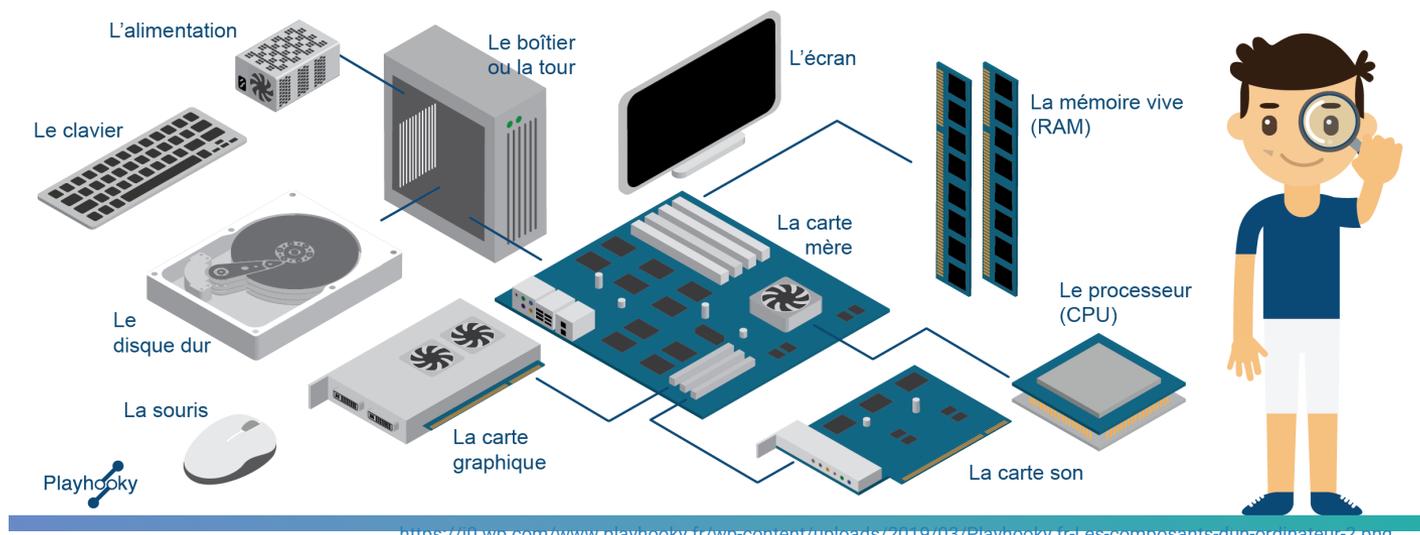
<https://edu.gcfglobal.org/en/computerbasics/inside-a-computer/1/>



conceptuel

# Informatique, ordinateur, système d'exploitation, application, application web, ... ?

- Composition



<https://10.wp.com/www.playnooky.fr/wp-content/uploads/2019/03/Playnooky.fr-Les-composants-dun-ordinateur-2.png>

- Une carte mère
  - Un processeur
  - Des prises d'extension
- Des composants de stockage
  - Mémoire vive
  - Disque dur
- Des cartes d'extension
  - Carte graphique
  - Carte son
  - ...
- Des Entrées/Sorties (clavier, souris, écran, haut parleur)

11101111110  
 1000010101  
 01010010100  
 00111111001  
 00000010110

Tout est stocké et transite en nombre binaire !

Ce sont des ressources (internes ou externes, utilisées par des instructions spécifiques du processeur)

# Informatique, ordinateur, système d'exploitation, application, application web, ... ?

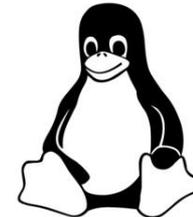
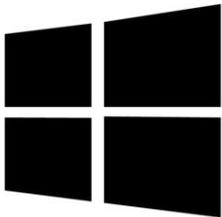
- Définitions

Un système d'exploitation, ou OS pour *Operating System*, définit un ensemble de programmes chargé d'établir une relation entre les différentes ressources matérielles, les applications et l'utilisateur.

Dans un ordinateur, le système d'exploitation a plusieurs missions. Il gère, entre autres, le processeur et la mémoire vive, optimise l'exécution des applications en leur attribuant les ressources nécessaires, fournit un certain nombre d'informations sur le bon fonctionnement de l'ordinateur, etc. Il permet l'utilisation de périphériques dans les meilleures conditions et protège l'accès aux ressources.

JDN  
JOURNAL DU NET

<https://www.journaldu.net/fr/web-tech/dictionnaire-du-webmastering/1203371-systeme-d-exploitation-definition-traduction-et-acteurs/>



# Informatique, ordinateur, système d'exploitation, application, application web, ... ?

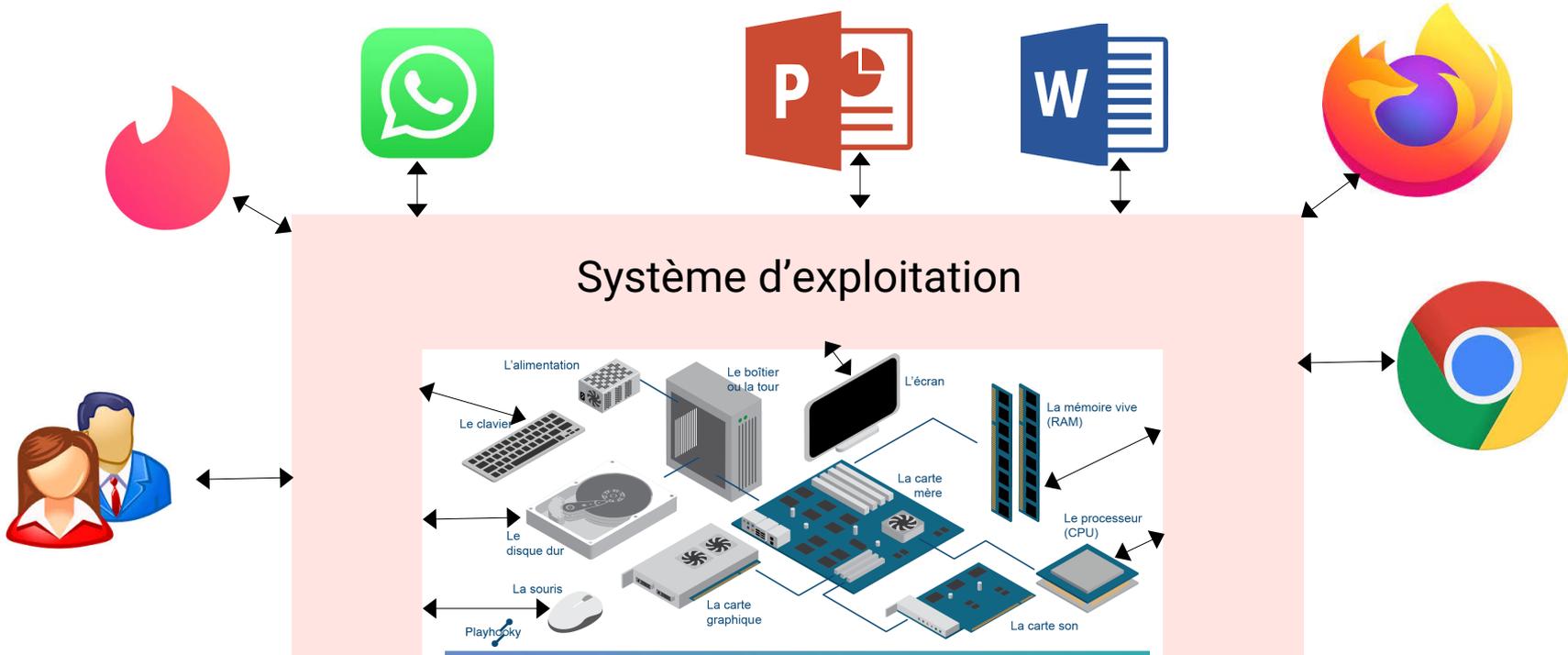
- Définitions

Un système d'exploitation, ou OS pour *Operating System*, définit un ensemble de programmes chargé d'établir une relation entre les différentes ressources matérielles, les applications et l'utilisateur.

Dans un ordinateur, le système d'exploitation a plusieurs missions. Il gère, entre autres, le **processeur** et la mémoire vive, optimise l'exécution des applications en leur attribuant les ressources nécessaires, fournit un certain nombre d'informations sur le bon fonctionnement de l'ordinateur, etc. Il permet l'utilisation de périphériques dans les meilleures conditions et protège l'accès aux ressources.

**JDN**  
JOURNAL DU NET

<https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203371-systeme-d-exploitation-definition-traduction-et-acteurs/>



# Informatique, ordinateur, système d'exploitation, application, application web, ... ?

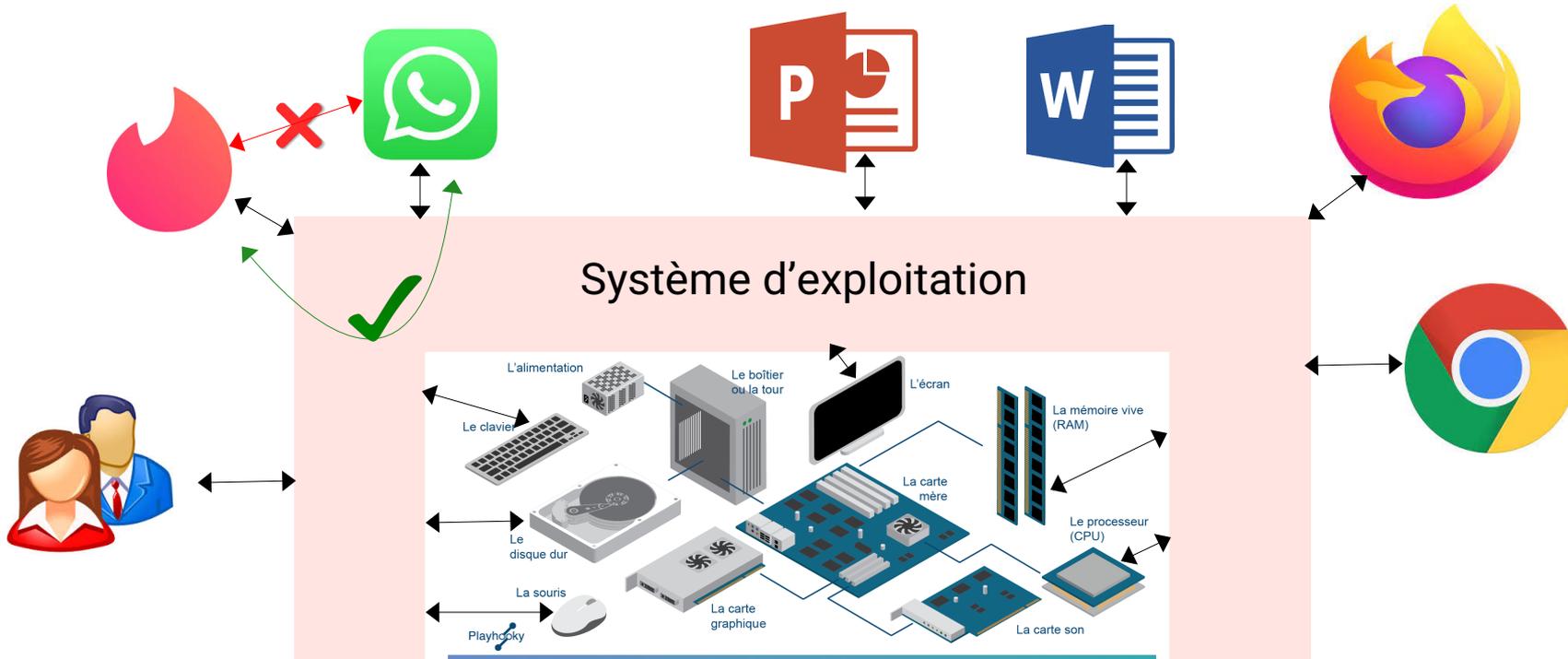
- Définitions

Un système d'exploitation, ou OS pour *Operating System*, définit un ensemble de programmes chargé d'établir une relation entre les différentes ressources matérielles, les applications et l'utilisateur.

Dans un ordinateur, le système d'exploitation a plusieurs missions. Il gère, entre autres, le processeur et la mémoire vive, optimise l'exécution des applications en leur attribuant les ressources nécessaires, fournit un certain nombre d'informations sur le bon fonctionnement de l'ordinateur, etc. Il permet l'utilisation de périphériques dans les meilleures conditions et protège l'accès aux ressources.

**JDN**  
JOURNAL DU NET

<https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203371-systeme-d-exploitation-definition-traduction-et-acteurs/>



# Informatique, ordinateur, système d'exploitation, application, application web, ... ?

- Definitions

## en Informatique

9. Domaine délimité pour le traitement duquel est écrit un **programme**.

10. **Programme ou ensemble de programmes** destiné à aider l'utilisateur d'un ordinateur pour le traitement d'une tâche précise.

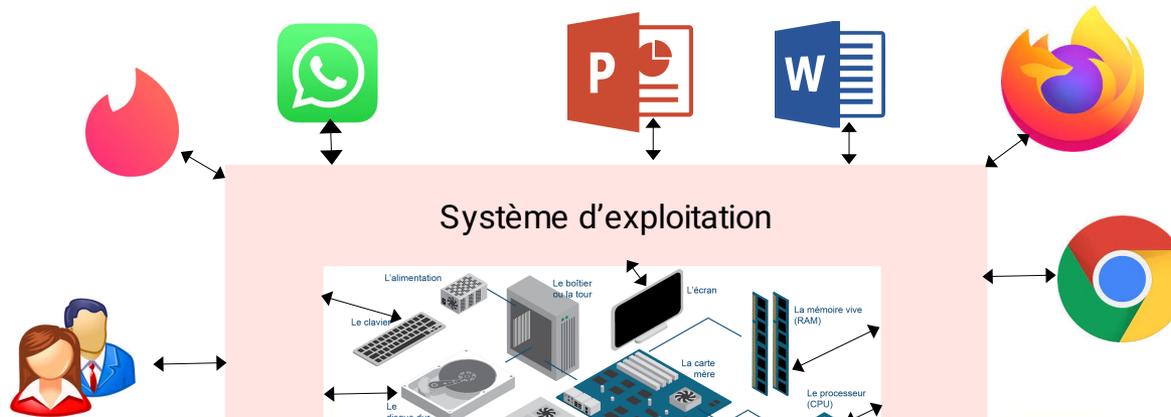


## Application

En informatique, une application est un **programme** utilisé pour réaliser une tâche ou un ensemble de tâches. Le terme est synonyme de logiciel. Un éditeur de texte, un jeu vidéo, un navigateur web sont des applications ; elles ont besoin des services d'un système d'exploitation pour fonctionner



<https://digitum.huma-num.fr/ressources/glossaire/>



# Informatique, ordinateur, système d'exploitation, application, application web, ... ?

## • Definitions

### Informatique

9. **Domaine délimité** pour le traitement duquel est écrit un **programme**.

10. **Programme ou ensemble de programmes** destiné à aider l'utilisateur d'un ordinateur pour le traitement d'une **tâche précise**.

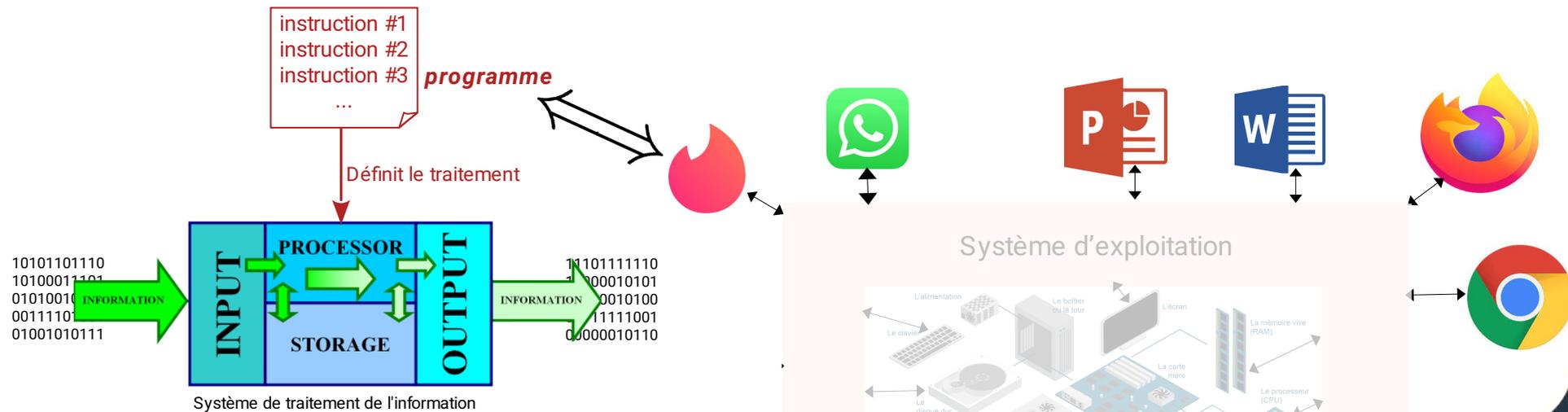


## Application

En informatique, une application est un **programme** utilisé pour réaliser une tâche ou un ensemble de tâches. Le terme est synonyme de logiciel. Un éditeur de texte, un jeu vidéo, un navigateur web sont des applications ; elles ont besoin des services d'un système d'exploitation pour fonctionner



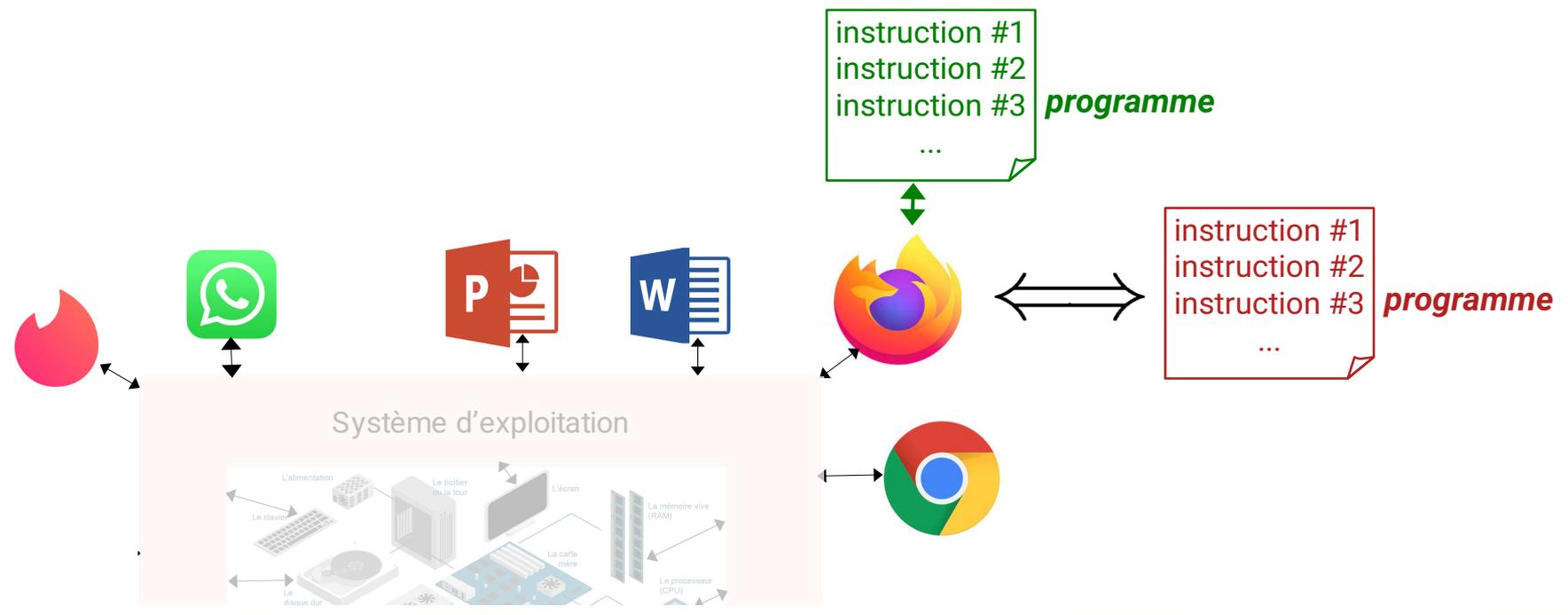
<https://dighum.huma-num.fr/ressources/glossaire/>



# Informatique, ordinateur, système d'exploitation, application, **application web**, ... ?

- Définitions

- C'est une application, et donc un programme, qui utilise les services d'un navigateur web plutôt que ceux d'un système d'exploitation.
- Elle fait également souvent appel aux services hébergés ailleurs et accessible via le réseau



# Informatique, ordinateur, système d'exploitation, application, application web, ... ?

- Définitions

## Informatique

9. **Domaine délimité** pour le traitement duquel est écrit un **programme**.

10. **Programme ou ensemble de programmes** destiné à aider l'utilisateur d'un ordinateur pour le traitement d'une **tâche précise**.



En informatique, une application est un **programme** utilisé pour réaliser une tâche ou un ensemble de tâches. Le terme est synonyme de logiciel. Un éditeur de texte, un jeu vidéo, un navigateur web sont des applications ; elles ont besoin des services d'un système d'exploitation pour fonctionner

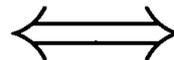


Trouvé sur <https://digithum.huma-num.fr/ressources/glossaire/>

C'est ça que l'on va faire ensemble dans cette UE. Écrire des programmes qui définissent une application de traitement de l'information pour un traitement donné

instruction #1  
instruction #2  
instruction #3  
...

**programme**



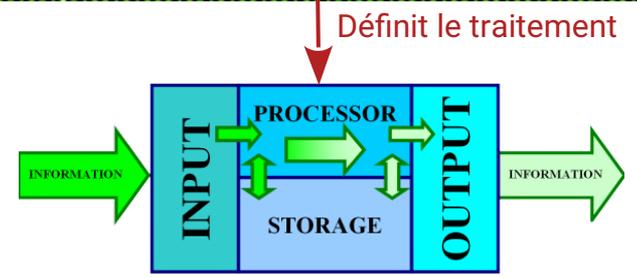
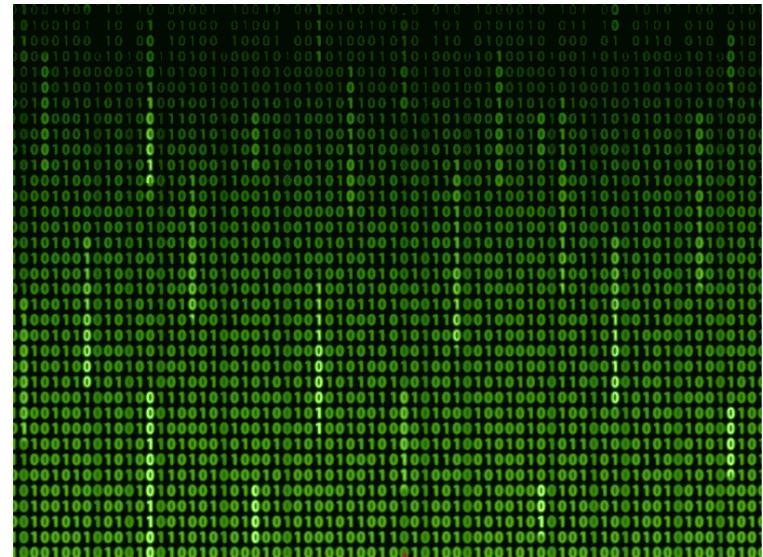
# Programmation ?

- Programmer c'est donner l'ordre des instructions qui représente le traitement que l'on veut faire sur les données d'entrées.
  - Instruction ?
    - Dans quel langage ?
    - À qui ?
  - Ordre des instructions ?
    - Et si je dois faire quelque chose 10000 fois ?
    - Et si ca dépend de ce que l'on me donne en entrée ?

# Programmation ?

- Programmer c'est donner l'ordre des instructions qui représentent le traitement que l'on veut faire sur les données d'entrées.
- Instruction ?
  - Dans quel langage ?
  - À qui ?

⇒ les instructions sont destinées au processeur de l'ordinateur. Le langage compris par le processeur est extrêmement simple, il ne comprend que des manipulations de base de l'information numérique ; et est lui même binaire...



# Programmation ?

- Programmer c'est donner l'ordre des instructions qui représente le traitement que l'on veut faire sur les données d'entrées.
- Instruction ?
  - Dans quel langage ?

*Différents styles et difficultés d'apprentissage ; largement basés sur l'anglais (niveau 6<sup>ème</sup>)*

⇒ on utilise des « **langages simples** » pour l'humain qui seront traduits automatiquement en binaire pour être donnés au processeur.



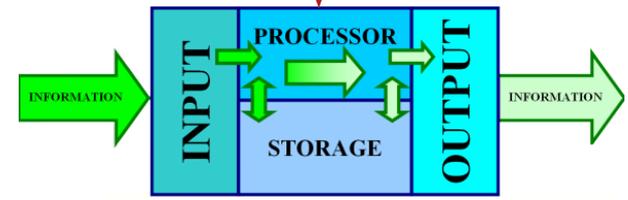
Une fois pour toute par un **compilateur** ou à la volée par un **interpréteur**

```
fichierSource = open("données.iiv")
for ligne in fichierSource_:
    print(int(ligne)*2)
fichierSource.close()
```

traduit



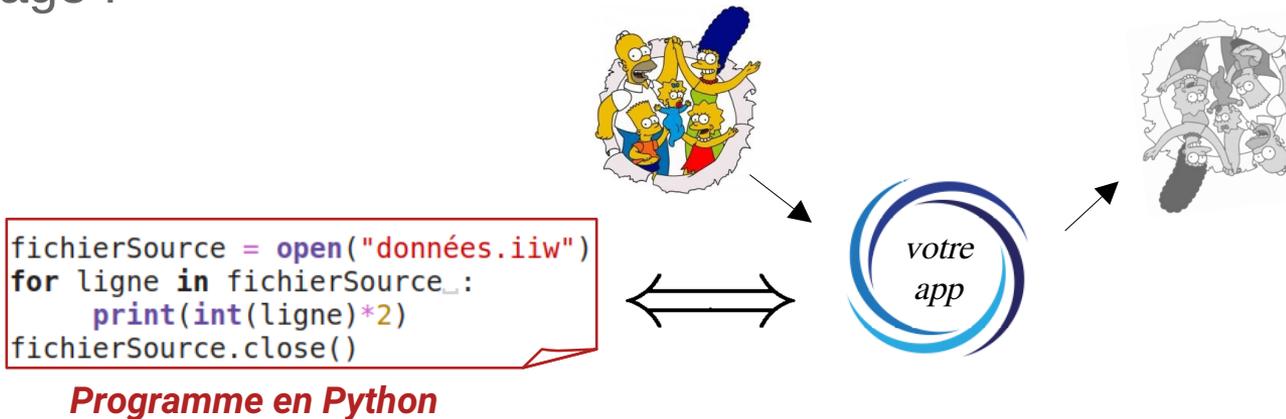
Définit le traitement





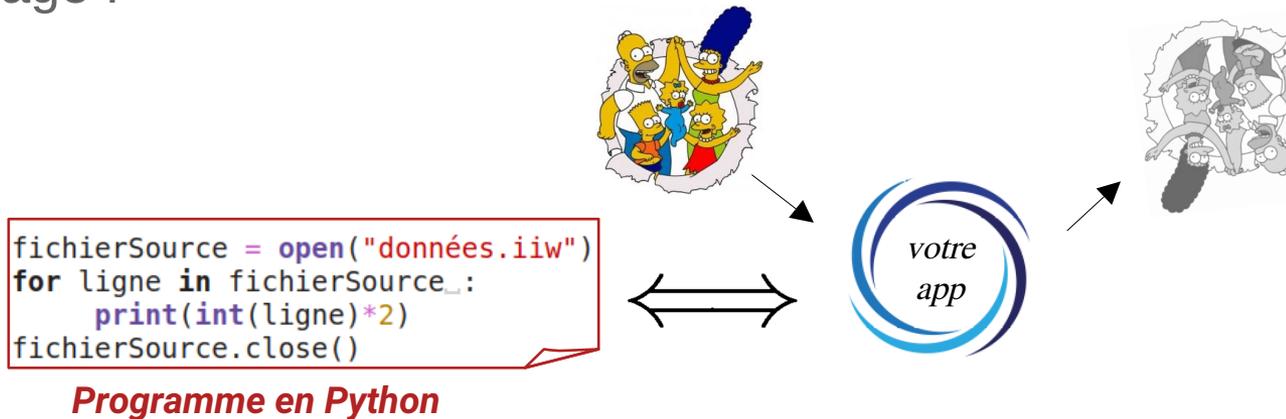
# Donc...

- ce que l'on va faire dans la première partie de l'UE c'est utiliser le langage informatique Python (version 3) pour définir une application de traitement de l'information permettant de faire de la manipulation d'image !



# Donc...

- ce que l'on va faire dans la première partie de l'UE c'est utiliser le langage informatique Python (version 3) pour définir une application de traitement de l'information permettant de faire de la manipulation d'image !



L'ordinateur est une machine électronique programmable capable de réaliser des calculs logiques **sur des nombres binaires.**

Donc, par exemple, un ordinateur ne peut traiter une image que si celle-ci est représentée comme une suite de nombre !!

⇒ on doit d'abord comprendre comment différentes informations peuvent être définies/structurées uniquement avec des nombres...

# Structuration de l'information

- Comment l'information est-elle stockée en terme de 0 et de 1 ?

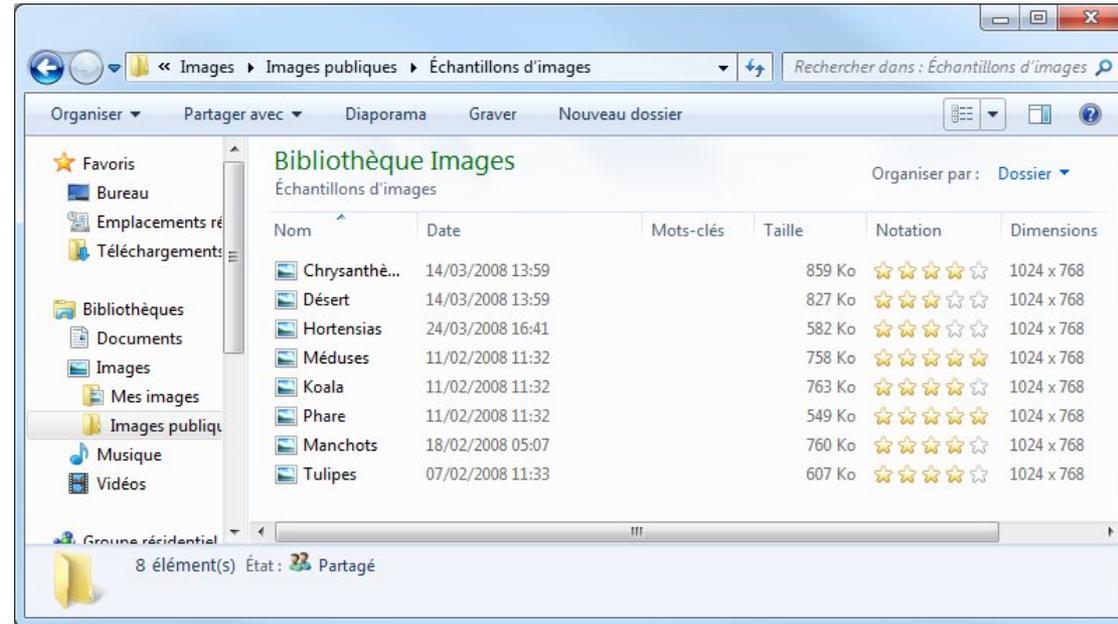
⇒ différencions 2 cas :

- 1) Le stockage sur le disque dur, i.e., ce que vous voyez dans le navigateur de fichier
- 2) Le stockage en mémoire vive ou « RAM » qui est créée par une application donnée

# Structuration de l'information

## Le stockage sur le disque dur

L'extension du fichier permet souvent de savoir comment est stockée l'information :  
*bmp, docx, csv, html, ...*

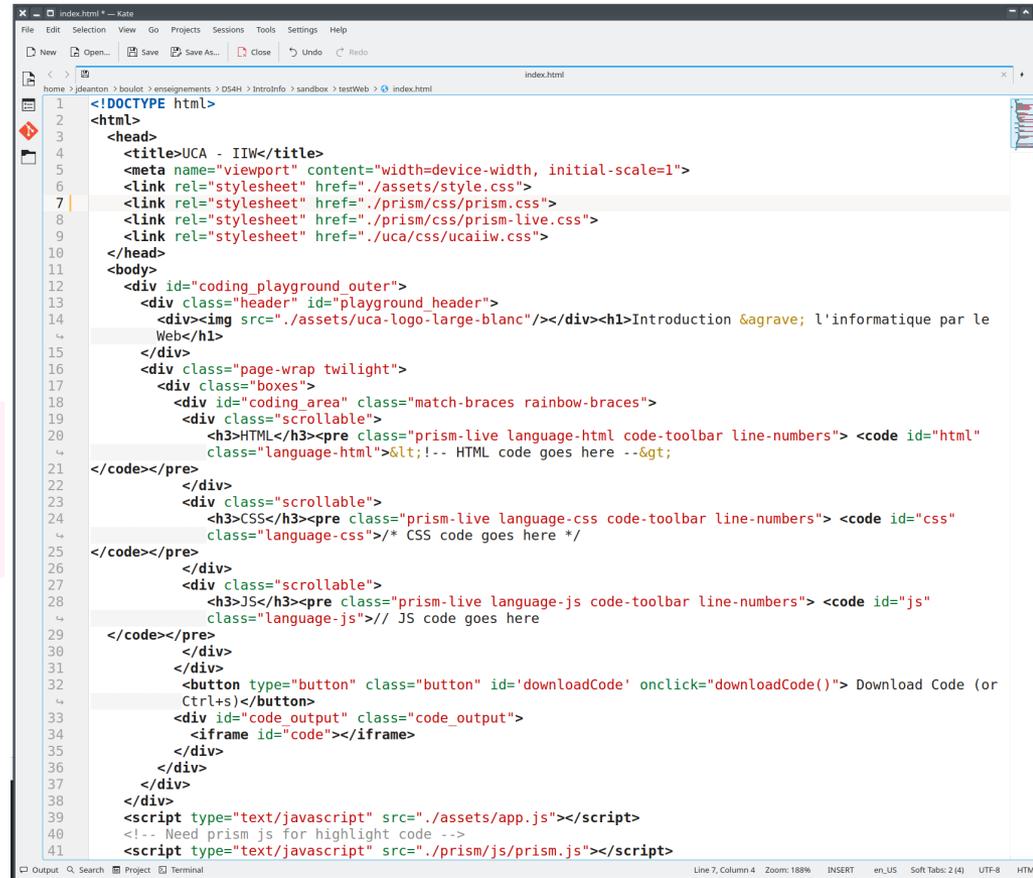


# Structuration de l'information

## Le stockage sur le disque dur

L'extension du fichier permet souvent de savoir comment est stockée l'information :  
 bmp, docx, csv, **html**, ...

⇒ ce sont des fichiers textes ;  
 on peut lire le contenu avec notepad ou équivalent.



```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>UCA - IIW</title>
5     <meta name="viewport" content="width=device-width, initial-scale=1">
6     <link rel="stylesheet" href="./assets/style.css">
7     <link rel="stylesheet" href="./prism/css/prism.css">
8     <link rel="stylesheet" href="./prism/css/prism-live.css">
9     <link rel="stylesheet" href="./uca/css/ucaiiv.css">
10  </head>
11  <body>
12    <div id="coding_playground_outer">
13      <div class="header" id="playground_header">
14        <div></div><h1>Introduction &agrave; l'informatique par le
15        Web</h1>
16      </div>
17      <div class="page-wrap twilight">
18        <div class="boxes">
19          <div id="coding_area" class="match-braces rainbow-braces">
20            <div class="scrollable">
21              <h3>HTML</h3><pre class="prism-live language-html code-toolbar line-numbers"> <code id="html"
22                class="language-html">&lt;&lt;!-- HTML code goes here --&gt;&gt;
23            </pre></div>
24          </div>
25          <div class="scrollable">
26            <h3>CSS</h3><pre class="prism-live language-css code-toolbar line-numbers"> <code id="css"
27              class="language-css">/* CSS code goes here */
28            </pre></div>
29          <div class="scrollable">
30            <h3>JS</h3><pre class="prism-live language-js code-toolbar line-numbers"> <code id="js"
31              class="language-js">// JS code goes here
32            </pre></div>
33          </div>
34          <button type="button" class="button" id="downloadCode" onclick="downloadCode()"> Download Code (or
35            Ctrl+s)</button>
36          <div id="code_output" class="code_output">
37            <iframe id="code"></iframe>
38          </div>
39          <script type="text/javascript" src="./assets/app.js"></script>
40          <!-- Need prism js for highlight code -->
41          <script type="text/javascript" src="./prism/js/prism.js"></script>
    
```

Fichier .html dans un éditeur de texte



# Structuration de l'information

## Fichier texte

⇒ un fichier texte est un fichier dont le contenu, en terme d'octet et donc de 0 et de 1, représente une suite de nombre, dont les valeurs représentent des caractères.

# 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]

Si on écrit dans un fichier texte en ASCII un octet dont la valeur est supérieure à 127 (0x7F), alors ce n'est pas un caractère !

# Structuration de l'information

## Fichier texte

⇒ un fichier texte est un fichier dont le contenu, en terme d'octet et donc de 0 et de 1, représente une suite de nombre, dont les valeurs représentent des caractères.

# 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]

Si on écrit dans un fichier texte en ASCII un octet dont la valeur est supérieure à 127 (0x7F), alors ce n'est pas un caractère !

Mais il existe plus de caractères qui sont spécifiques à différentes langues !

# Structuration de l'information

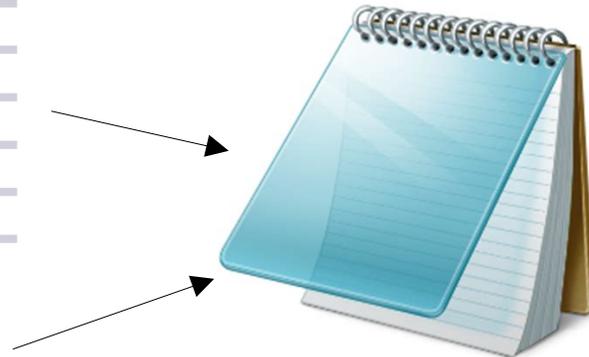
## Fichier texte

⇒ un fichier texte est un fichier dont le contenu, en terme d'octet et donc de 0 et de 1, représente une suite de nombre, dont les valeurs représentent des caractères.

### ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NUL]	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	[END 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]

10101101 11010100  
 01110101 01001010  
 00011110 10010100  
 10101110 ...



Bonjour !

Une application permettant d'ouvrir un fichier texte lit un ensemble d'octets et regarde dans la table renseignée par l'utilisateur (configuré par défaut par là où il habite) quels sont les caractères à afficher

# Structuration de l'information

- Comment l'information est-elle stockée en terme de 0 et de 1 ?

⇒ différencions 2 cas :

- 1) Le stockage sur le disque dur, i.e., ce que vous voyez dans le navigateur de fichier
- 2) **Le stockage en « RAM » qui est crée par une application donnée**

# Structuration de l'information

## Variable

- L'information en RAM est accessible via une **variable**. C'est donc un emplacement mémoire dont on peut accéder à la valeur en utilisant le nom de la variable.
- Sa structure dépend en partie du langage utilisé et de la volonté du développeur. La structure est conditionnée par l'usage.
- Elles permettent de stocker de l'information utile pour le programme. Par exemple dans le langage Python on peut **déclarer** (i.e., créer) des variables comme ceci :

```
i = 42 # variable nommée i et de valeur 42
firstname = "Julien" # variable nommée firstname et de valeur Julien
```

- En mémoire nous aurons donc un octet contenant la valeur 0b0010 1010 (i.e., 42 en base 2) et un 6 octets contenant 0x4A 75 6C 69 65 6E (i.e., Julien en code ASCII).
- ⇒ la bonne nouvelle c'est que toute les conversions sont transparentes pour la personne qui programme

# Structuration de l'information

## Type des Variables

- Elles permettent de stocker de l'information utile pour le programme. Par exemple dans le langage Python on peut **déclarer** (i.e., créer) des variables comme ceci :

```
i = 42 # variable nommée i et de valeur 42
firstname = "Julien" # variable nommée firstname et de valeur Julien
```

- Dans les exemples précédents, le type de la variable est implicite et inféré (i.e., déduit de la valeur à droite du signe égal). Selon vous de quel type est `i` ? Quid de `firstname` ?

# Structuration de l'information

## Type des Variables

- Elles permettent de stocker de l'information utile pour le programme. Par exemple dans le langage Python on peut **déclarer** (i.e., créer) des variables comme ceci :

```
i = 42 # variable nommée i et de valeur 42  
firstname = "Julien" # variable nommée firstname et de valeur Julien
```

- Dans les exemples précédents, le type de la variable est implicite et inféré (i.e., déduit de la valeur à droite du signe égal). Selon vous de quel type est `i` ? Quid de `firstname` ?
  - `i` est de type *Integer* ou *int* (i.e., entier naturel) et `firstname` est de type *String* ou *str*, (i.e., chaîne de caractères).

# Structuration de l'information

## Type des Variables

- Rôle d'un type de donnée :
  - Permettre à l'ordinateur de savoir comment interpréter le nombre binaire enregistré en mémoire
  - Permettre de savoir quelles sont les *opérations* que l'on sait faire sur une variable
  - S'assurer que l'on ne mélange pas des choux et des carottes :

```
i = 42           # variable nommée i et de valeur 42
firstname = "Julien" # variable nommée firstname et de valeur Julien
firstname + 1     # TypeError:
                  # can only concatenate str (not "int") to str
```

# Structuration de l'information

## Type des Variables : bonnes pratiques

- spécifier le type d'une variable explicitement. Par exemple dans le langage Python :

```
i : int = int(42) # variable nommée i et de valeur 42
firstname : str = str("Julien") # variable nommée firstname et de valeur Julien
```

- `int(42)` est un appel à la *fonction d'initialisation* du type `int` avec la valeur 42
- Nommer vos variables correctement afin de vous rappeler la signification de la valeur qu'elle contient.

 Dans le langage Python une variable peut changer de type dynamiquement (ce n'est pas le cas de tous les langages). Éviter de le faire (on en reparlera plus tard).

# Structuration de l'information

## Types prédéfinis (*~Built-in*)

```
i : int = int(42)           # variable nommée i et de valeur 42
firstname : str = str("Julien") # variable nommée firstname et de valeur Julien
```

⇒ dans les exemples ci dessus les types entier naturel et chaîne de caractère existent de base dans le langage Python. Ces types sont appelés **types prédéfinis** ; i.e., ce sont les types « *de base* » qui existent dans le langage. Voici certains des types prédéfinis du langage Python :

- Integer ⇒  $\mathbb{N}$
- Float ⇒  $\mathbb{R}$  (warning)
- String ⇒ chaîne de caractères
- Boolean ⇒ Vrai ou Faux

⇒ Python va automatiquement mettre les nombres binaires correspondants à l'encodage du type dans la mémoire

# Structuration de l'information

## *Opération sur les variables (1/3)*

- On peut réaliser les opérations sur les variables telles que définies par le type. Par exemple en Python
- ⚠ vous allez voir des instructions qui vont s'exécuter l'une après l'autre sur le processeur. L'ordre dans lequel elles s'exécutent est important

# Structuration de l'information

## Opération sur les variables (1/3)

- On peut réaliser les opérations sur les variables telles que définis par le type. Par exemple en Python

***nomVar : Type = initialization value***

```
i : int = int(42)    #déclaration de la variable i, initialisée à 42
j : int = 2         #déclaration de la variable j, initialisée à 2
k : int = i+j      #déclaration de la variable k, initialisée à 44
                  # appel de l'opération addition avec pour paramètres i et j
l : int = k//2     #déclaration de la variable l, initialisée à 22
                  # appel de l'opération division entière avec pour
                  # paramètres k et 2

s1 : str = str("Hello")
s2 : str = str(" ")
s3 : str = "world"
greeting : str = s1+s2+s3    #déclaration de la variable greeting
                            # initialisée à Hello World
                            # 2 appels de l'opération concaténation avec pour
                            # paramètres s1 et s2 puis le résultat de s1+s2 et s3
```

# Structuration de l'information

## Opération sur les variables (1/3)

- On peut réaliser les opérations sur les variables telles que définis par le type. Par exemple en Python

```
i : int = int(42)
j : int = 2
k: int = i+j
```

```
l: int = k//2
```

```
s1: str = str("Hello")
s2: str = str(" ")
s3: str = "world"
```

```
greeting: str = s1+s2+s3      #déclaration de la variable greeting
                               # initialisée à Hello World
                               # 2 appels de l'opération concaténation avec pour
                               # paramètres s1 et s2 puis le résultat de s1+s2 et s3
```

### C **concaténer**

Verbe

#### DÉFINITION

**concaténer**, *verbe*

Voir aussi [concaténation](#).

Sens 1 [Informatique](#) Du latin "cum" (ensemble) et "catena" (chaîne), "concaténer" est l'action de [relier deux chaînes informatiques](#) pour en [créer une nouvelle](#).

# Structuration de l'information

## Opération sur les variables (1/3)

- On peut réaliser les opérations sur les variables telles que définis par le type. Par exemple en Python

```
i : int = int(42)    #déclaration de la variable i, initialisée à 42
j : int = 2         #déclaration de la variable j, initialisée à 2
k: int = i+j       #déclaration de la variable k, initialisée à 44
                   # appel de l'opération addition avec pour paramètres i et j
del j              #suppression de la variable j de la mémoire

k = j+1            → erreur
```

*Les variables sont stockées dans la RAM. Même si on le fait peu, on peut à tout moment enlever une variable de la mémoire en utilisant **del***

# Structuration de l'information

## Opération sur les variables (2/3)

- On peut réaliser les opérations sur les variables telles que définis par le type. Par exemple en Python

```
i: int = int(42) # déclaration de la variable i, initialisée à 42
j: int = 2      # déclaration de la variable j, initialisée à 2
j = 15         # affectation de la valeur 15 à la variable j
```

-  l'opération d'affectation, représentée par le symbole = change la valeur de la variable. Elle est à différencier de l'égalité mathématique ! Elle prend la valeur de droite et remplace la valeur existante dans la variable à gauche

```
i: int = int(42)
j: int = int(2)
j = i
i = 6
```

Quelle est la valeur de j ?

# Structuration de l'information

## Opération sur les variables (3/3)

- On peut réaliser les opérations sur les variables telles que définis par le type. Par exemple en Python

Pour information, la liste de toutes les opérations possibles est consultable ici :

[https://www.w3schools.com/python/python\\_operators.asp](https://www.w3schools.com/python/python_operators.asp)

```
i: int = int(42)      # déclaration de la variable i, initialisée à 42
j: int = int(2)       # déclaration de la variable j, initialisée à 2
j = i - j             # affectation de la valeur 15 à la variable j

b1: bool = bool( i < j ) # comparaison entre deux entier.
                        # Renvoie vrai ou faux

b2: bool = bool( i == j )
b3: bool = bool(not b1)

f : float = float(j / i) # approximation de la valeur de j / i
                        # stockée dans un float
                        # attention ne jamais comparer 2 float avec une égalité (==)

f1 = i**2             # i puissance 2 (au carré)
f2 : float = f1**(1/2) # j puissance 1/2 (racine carrée de j)
```

# Structuration de l'information

## Liste

Afin de stocker plusieurs éléments dans une seule variable, on utilise un type *Collection*. Il existe différentes collections mais nous ne verrons dans ce cours que le type prédéfini «*list*». En python, une liste est :

- Ordonnée et indexable. Les éléments sont rentrés et stockés dans un ordre particulier et on peut facilement récupérer l'élément à une position particulière.

```
mesAmis: list[str] = ['tata', 'titi', 'toto'] # une variable de type list
ami0: str = mesAmis[0] # une variable de type string initialisée avec l'élément à l'indice 0, i.e., 'zaza'
titi: str = mesAmis[1] # attention titi est le nom de la variable, et 'titi' est son contenu
```

- Modifiable. Il est possible de modifier, d'ajouter ou d'enlever des éléments.

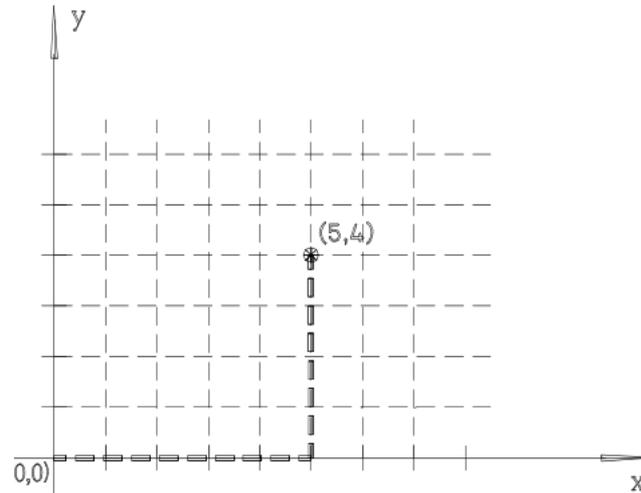
```
mesAmis[0] = 'zaza' # ['zaza', 'titi', 'toto']
mesAmis = mesAmis + ['zozo'] # ['zaza', 'titi', 'toto', 'zozo']
del mesAmis[1] # ['zaza', 'toto', 'zozo']
mesAmis = mesAmis + 'zozo' # TypeError: can only concatenate list (not "str") to list
```

- Peut contenir plusieurs fois le même élément

# Structuration de l'information

*Types structurés, définis par le développeur*

- Comment avoir un type qui représente un point dans un repère 2D orthonormé, dont les valeurs en abscisse et en ordonnée sont des entiers naturels ?



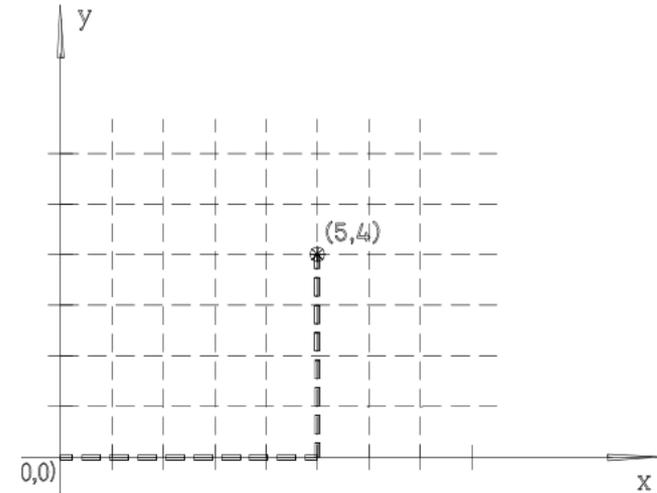
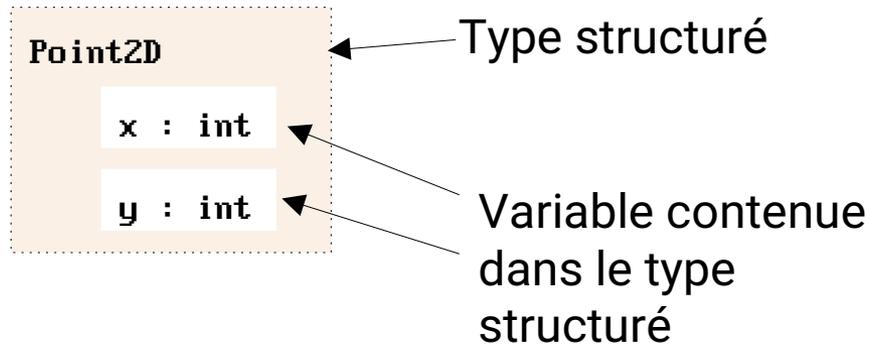
```
p1 : Point2D = Point2D (5,4) #?
```

- Plus précisément on voudrait un **type personnalisé Point2D** pour stocker les valeurs de l'abscisse et de l'ordonnée et que l'on puisse facilement accéder à l'abscisse en la nommant `x` et à l'ordonnée en la nommant `y` afin de pouvoir déclarer des variables comme `p1` ci dessus.

# Structuration de l'information

*Types structurés, définis par le développeur*

- On définit **un nouveau type** comme une structure hiérarchique contenant des variables d'un type primitif (ou non)



- En définissant cette structure on peut :
  - définir des variables de ce type :
    - `p1 : Point2D = Point2D(5,4)`
    - `p2 : Point2D = Point2D(x=5,y=4)`
- accéder à la valeur des variables internes en écrivant : `p1.x` ou bien `p1.y`

exemple :

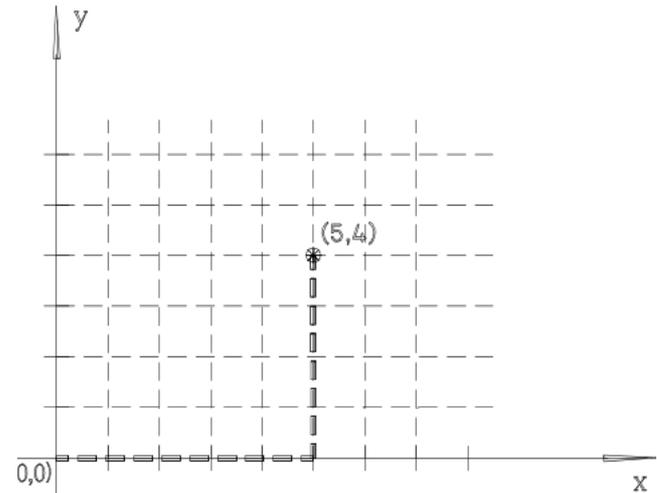
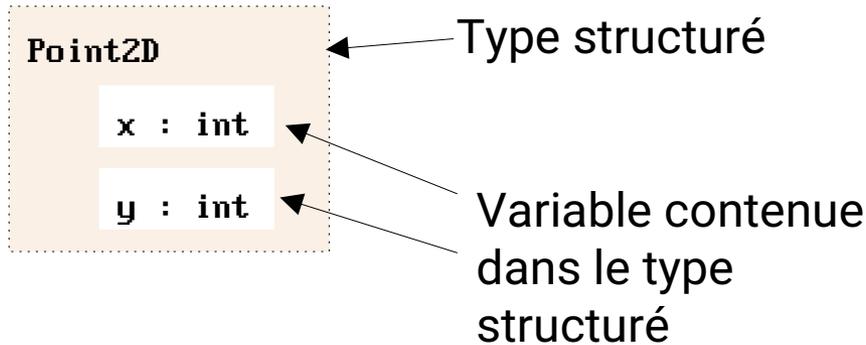
```
p1.x = 6
```

```
p1.y = p1.y // 2
```

# Structuration de l'information

*Types structurés, définis par le développeur*

On définit **un nouveau type** comme une structure hiérarchique contenant des variables d'un type primitif (ou non)



```

from dataclasses import dataclass           #required to declare a dataclass

@dataclass
class Point2D:
    x : int
    y : int

p1 : Point2D = Point2D (5,4) #variables internes initialisées dans l'ordre
                               p1.x = 5, p1.y = 4
    
```

# Structuration de l'information

*Types structurés, définis par le développeur*

On définit **un nouveau type** comme une structure hiérarchique contenant des variables d'un type primitif (ou non)

```
from dataclasses import dataclass          #required to declare a dataclass
```

```
@dataclass ← Annotation pour se simplifier la  
class Point2D: ← Mot clé class suivi du nom du type et de ' '  
    x : int | ← Ensemble de variables internes typées et ordonnées  
    y : int
```

```
p1 : Point2D = Point2D (5,4) #variables internes initialisées dans l'ordre  
                               p1.x = 5, p1.y = 4
```

```
p2 : Point2D = Point2D (10,2)
```

```
values: list[Point2D] = [p1, p2]          #[Point2D(x=5,y=4), Point2D(x=10,y=2)]
```

# Résumé

- On veut traiter de l'information automatiquement tel que défini par un programme.
  - Un programme est une suite ordonnée d'instructions
  - Tout dans l'ordinateur est stocké à l'aide de nombres binaires.
  - Il est nécessaire de comprendre comment est structurée l'information en terme de variables et de leur type.
  - Un langage informatique offre des facilités de manipulation des variables et des types (opérations et types prédéfinis, facilité de création de nouveaux types structurés, ...)
- ⇒ Pour l'instant toutes nos instructions s'exécutent toujours dans l'ordre donné. Le prochain challenge consiste à définir des ordres conditionnels pour pouvoir définir des algorithmes intéressants.