

Algorithmique de base et python

Cours 04

Master G2M

2024-2025

celine.jost@univ-paris8.fr

*Reformuler,
interpréter, vérifier
et écrire*

*Opérateurs
Concaténation*

*Condition
Imbrication*

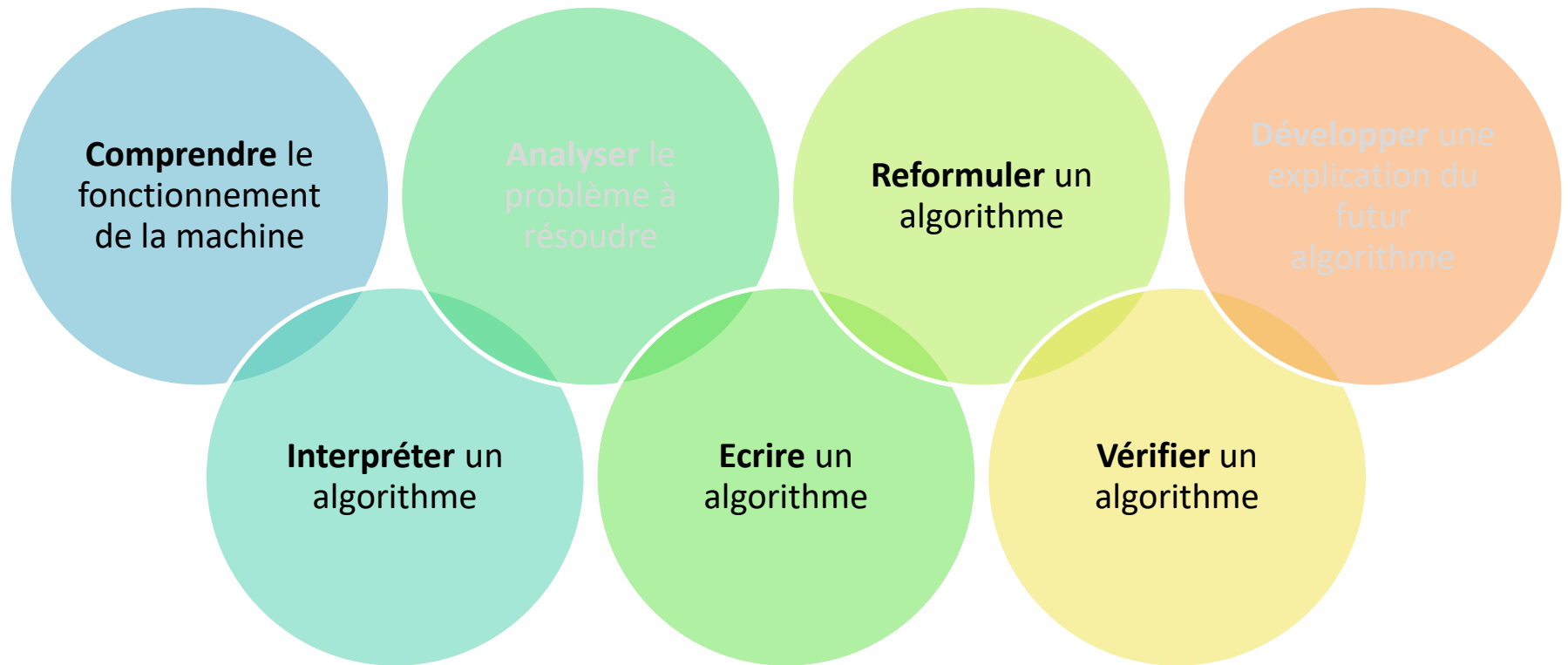
Bloc d'instructions



```
debut
  imc <- 27.0
  resultat <- ""
  si imc < 18.5 alors
    resultat <- "sous-poids"
  sinon si imc < 25.0 alors
    resultat <- "normal"
  sinon si imc < 30.0 alors
    resultat <- "surpoids"
  sinon si imc < 35.0 alors
    resultat <- "obésité"
  sinon
    resultat <- "obésité sévère"
  finsi
fin
```



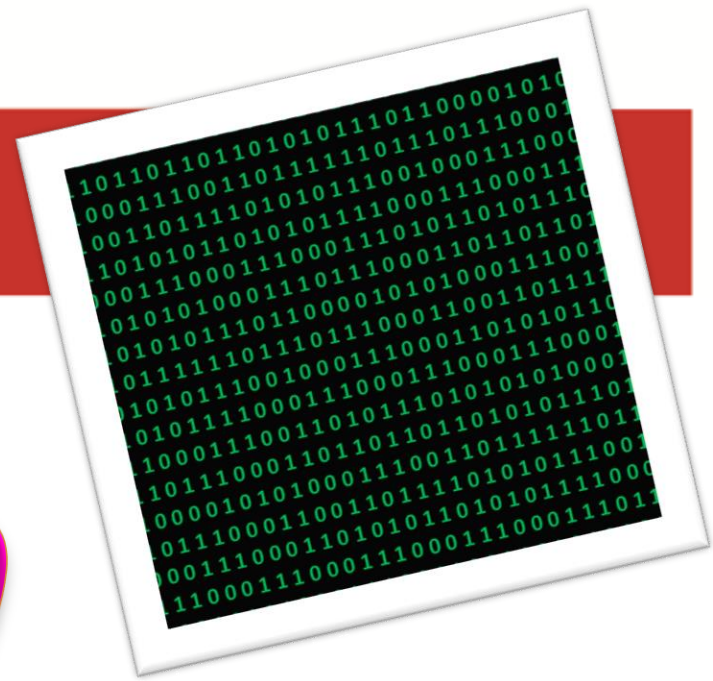
Les savoir-faire



Ce n'est pas une définition officielle. Cela constitue mes choix pédagogiques.



- ~~Constante~~
- ~~Variable~~
- Opérateur
- *Tableau (liste, pile)*
- *Arbre*
- ~~Séquence~~
- Condition
- *Itération*
- *Fonction*
- *Procédure*
- ~~Permutation~~
- Concaténation
- *Tri*
- ~~Déclaration~~
- ~~Affectation~~
- ~~Initialisation~~
- *Incrémentation*
- ~~Expression~~
- ~~Instruction~~
- Bloc d'instructions
- ~~Lecture~~
- ~~Ecriture~~
- Imbrication
- *Récursivité*
- *Complexité*



0 0 1 0 1 0 0

LES CONDITIONS



Règles de calcul des impôts



- Si la personne est au SMIC, alors prélever 10% de son salaire.
- Si la personne est de classe moyenne, prélever 25% de son salaire.
- Si la personne est riche, ne surtout pas toucher son argent.



Règles de calcul des impôts

- **Si la personne est au SMIC**, alors prélever 10% de son salaire.
- **Si la personne est de classe moyenne**, prélever 25% de son salaire.
- **Si la personne est riche**, ne surtout pas toucher son argent.

Cela s'appelle des conditions.



Exemple 2

```
debut
  si age > 20 alors
    ecrire « C'est un sénior »
  sinon si age > 17 alors
    ecrire « C'est un junior »
  sinon si age > 15 alors
    ecrire « C'est un cadet »
  sinon si age > 13 alors
    ecrire « C'est un minime »
  sinon si age > 11 alors
    ecrire « C'est un benjamin »
  finsi
fin
```

Reformulation ?



L'exemple 2 est un algorithme qui différencie le traitement à effectuer en fonction de la catégorie sportive d'une personne, selon les critères suivants :

Senior : 21 ans et +
Junior : 18,19,20 ans
Cadet : 16,17 ans
Minime : 14,15 ans
Benjamin : 12,13 ans
Poussin : 9,10,11 ans
Poucet : 7,8 ans



Interprétation
un seul choix possible

```
age <- 16
si age > 20 alors
    // traitement senior
sinon si age > 17 alors
    // traitement junior
sinon si age > 15 alors
    /* traitement
       cadet */
sinon si age > 13 alors
    // traitement minime
sinon si age > 11 alors
    // traitement benjamin
sinon si ...

finsi
```

Avant d'interpréter cet
algorithme, intéressons
nous à : **//**.



Interprétation un seul choix possible

```
age <- 16
si age > 20 alors
    // traitement senior
sinon si age > 17 alors
    // traitement junior
sinon si age > 15 alors
    /* traitement
       cadet */
sinon si age > 13 alors
    // traitement minime
sinon si age > 11 alors
    // traitement benjamin
sinon si ...

finsi
```

Avant d'interpréter cet algorithme, intéressons nous à : **//**.

Cela s'appelle un **commentaire**. Tout ce qui suit ce symbole ne doit pas être interprété.

Il y a aussi **/*** et ***/**



Interprétation

```
age <- 16
si age > 20 alors
    // traitement senior
sinon si age > 17 alors
    // traitement junior
sinon si age > 15 alors
    // traitement cadet
sinon si age > 13 alors
    // traitement minime
sinon si age > 11 alors
    // traitement benjamin
sinon si ...

finsi
```

Rappel

Senior : 21 ans et +
Junior : 18,19,20 ans
Cadet : 16,17 ans
Minime : 14,15 ans
Benjamin : 12,13 ans
Poussin : 9,10,11 ans
Poucet : 7,8 ans



Interprétation

```
age <- 16
si age > 20 alors    ←
    // traitement senior
sinon si age > 17 alors
    // traitement junior
sinon si age > 15 alors
    // traitement cadet
sinon si age > 13 alors
    // traitement minime
sinon si age > 11 alors
    // traitement benjamin
sinon si ...

finsi
```

Est-ce que l'âge est supérieur à 20 ?

Rappel

Senior : 21 ans et +
Junior : 18,19,20 ans
Cadet : 16,17 ans
Minime : 14,15 ans
Benjamin : 12,13 ans
Poussin : 9,10,11 ans
Poucet : 7,8 ans



Interprétation

```
age <- 16
si age > 20 alors ←
    // traitement senior
sinon si age > 17 alors ←
    // traitement junior
sinon si age > 15 alors
    // traitement cadet
sinon si age > 13 alors
    // traitement minime
sinon si age > 11 alors
    // traitement benjamin
sinon si ...

finsi
```

Est-ce que l'âge est supérieur à 20 ?

NON

Est-ce que l'âge est supérieur à 17 ?

Rappel

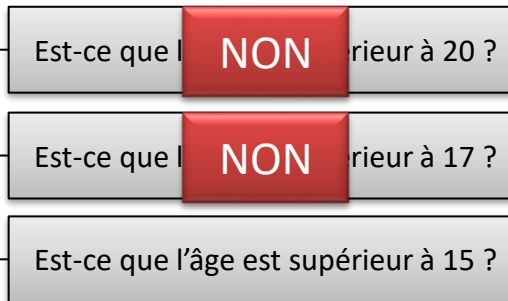
Senior : 21 ans et +
Junior : 18,19,20 ans
Cadet : 16,17 ans
Minime : 14,15 ans
Benjamin : 12,13 ans
Poussin : 9,10,11 ans
Poucet : 7,8 ans



Interprétation

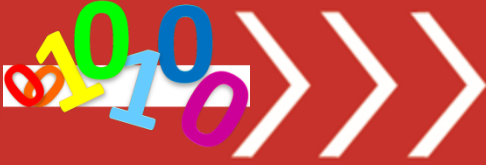
```
age <- 16
si age > 20 alors ←
    // traitement senior
sinon si age > 17 alors ←
    // traitement junior
sinon si age > 15 alors ←
    // traitement cadet
sinon si age > 13 alors
    // traitement minime
sinon si age > 11 alors
    // traitement benjamin
sinon si ...

finsi
```



Rappel

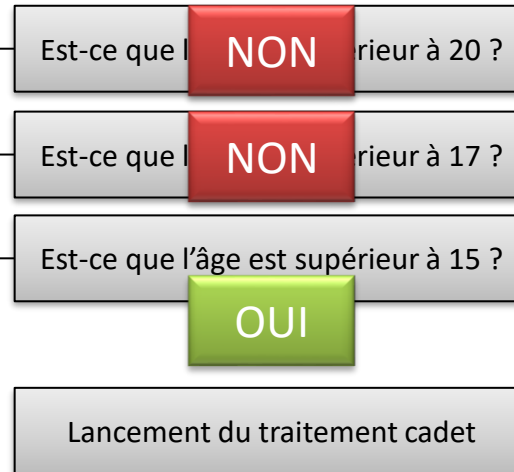
Senior : 21 ans et +
 Junior : 18,19,20 ans
 Cadet : 16,17 ans
 Minime : 14,15 ans
 Benjamin : 12,13 ans
 Poussin : 9,10,11 ans
 Poucet : 7,8 ans



Interprétation

```
age <- 16
si age > 20 alors ←
    // traitement senior
sinon si age > 17 alors ←
    // traitement junior
sinon si age > 15 alors ←
    // traitement cadet
sinon si age > 13 alors
    // traitement minime
sinon si age > 11 alors
    // traitement benjamin
sinon si ...

finsi
```



Rappel

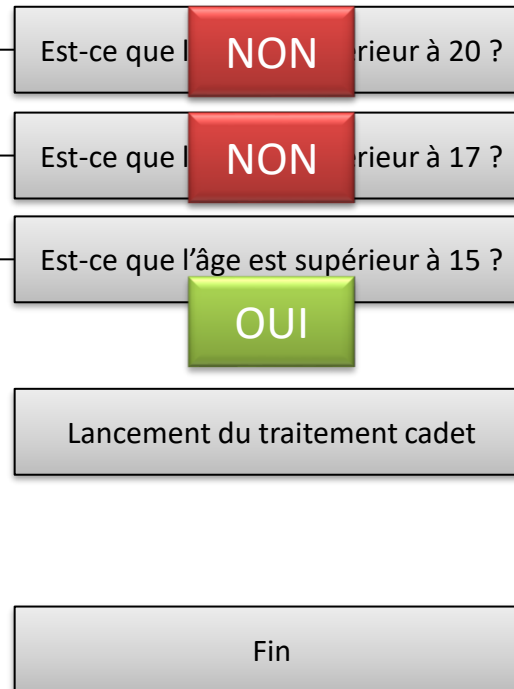
Senior : 21 ans et +
 Junior : 18,19,20 ans
 Cadet : 16,17 ans
 Minime : 14,15 ans
 Benjamin : 12,13 ans
 Poussin : 9,10,11 ans
 Poucet : 7,8 ans



Interprétation

```
age <- 16
si age > 20 alors ←
    // traitement senior
sinon si age > 17 alors ←
    // traitement junior
sinon si age > 15 alors ←
    // traitement cadet
sinon si age > 13 alors
    // traitement minime
sinon si age > 11 alors
    // traitement benjamin
sinon si ...

finsi
```



Rappel

Senior : 21 ans et +
 Junior : 18,19,20 ans
 Cadet : 16,17 ans
 Minime : 14,15 ans
 Benjamin : 12,13 ans
 Poussin : 9,10,11 ans
 Poucet : 7,8 ans



Un seul choix possible

Quelle est votre couleur préférée ?

- ☐ Jaune
- ☐ Vert
- ☐ Bleu
- ☐ Rouge
- ☐ Marron

Plusieurs choix possibles

Quels UE suivez-vous ?

- ☐ Anglais
- ☐ Allemand
- ☐ Français
- ☐ Chinois
- ☐ Russe



Choix unique vs plusieurs choix

Un seul choix possible

```
age <- 16
si age > 20 alors
    // traitement senior
sinon si age > 17 alors
    // traitement senior
sinon si age > 15 alors
    // traitement cadet
sinon si age > 13 alors
    // traitement minime
sinon si age > 11 alors
    // traitement benjamin
sinon si ...

finsi
```

Plusieurs choix possibles

```
age <- 16
si age > 20 alors
    // traitement senior
finsi
si age > 17 alors
    // traitement senior
finsi
si age > 15 alors
    // traitement cadet
finsi
si age > 13 alors
    // traitement minime
finsi
...
```



Interprétation Plusieurs choix possibles

```
age <- 16
si age > 20 alors
    // traitement senior
finsi
si age > 17 alors
    // traitement junior
finsi
si age > 15 alors
    // traitement cadet
finsi
si age > 13 alors
    // traitement minime
finsi
...
```

Rappel

Senior : 21 ans et +
Junior : 18,19,20 ans
Cadet : 16,17 ans
Minime : 14,15 ans
Benjamin : 12,13 ans
Poussin : 9,10,11 ans
Poucet : 7,8 ans



Interprétation Plusieurs choix possibles

```
age <- 16
si age > 20 alors    ←
    // traitement senior
finsi
si age > 17 alors
    // traitement junior
finsi
si age > 15 alors
    // traitement cadet
finsi
si age > 13 alors
    // traitement minime
finsi
...
```

Est-ce que l'âge est supérieur à 20 ?

Rappel

Senior : 21 ans et +
Junior : 18,19,20 ans
Cadet : 16,17 ans
Minime : 14,15 ans
Benjamin : 12,13 ans
Poussin : 9,10,11 ans
Poucet : 7,8 ans



Interprétation Plusieurs choix possibles

```
age <- 16
si age > 20 alors    ←
    // traitement senior
finsi
si age > 17 alors    ←
    // traitement junior
finsi
si age > 15 alors
    // traitement cadet
finsi
si age > 13 alors
    // traitement minime
finsi
...
```

Est-ce que l'âge est supérieur à 20 ?

NON

Est-ce que l'âge est supérieur à 17 ?

Rappel

Senior : 21 ans et +
Junior : 18,19,20 ans
Cadet : 16,17 ans
Minime : 14,15 ans
Benjamin : 12,13 ans
Poussin : 9,10,11 ans
Poucet : 7,8 ans



Interprétation Plusieurs choix possibles

```
age <- 16
si age > 20 alors ←
    // traitement senior
finsi
si age > 17 alors ←
    // traitement junior
finsi
si age > 15 alors ←
    // traitement cadet
finsi
si age > 13 alors
    // traitement minime
finsi
...
```

Est-ce que l'âge est supérieur à 20 ?

NON

Est-ce que l'âge est supérieur à 17 ?

NON

Est-ce que l'âge est supérieur à 15 ?

Rappel

Senior : 21 ans et +
Junior : 18,19,20 ans
Cadet : 16,17 ans
Minime : 14,15 ans
Benjamin : 12,13 ans
Poussin : 9,10,11 ans
Poucet : 7,8 ans



Interprétation Plusieurs choix possibles

```
age <- 16
si age > 20 alors ←
    // traitement senior
finsi
si age > 17 alors ←
    // traitement junior
finsi
si age > 15 alors ←
    // traitement cadet
finsi
si age > 13 alors
    // traitement minime
finsi
...
```

Est-ce que l'âge est supérieur à 20 ? **NON**

Est-ce que l'âge est supérieur à 17 ? **NON**

Est-ce que l'âge est supérieur à 15 ? **OUI**

Rappel

Senior : 21 ans et +
Junior : 18,19,20 ans
Cadet : 16,17 ans
Minime : 14,15 ans
Benjamin : 12,13 ans
Poussin : 9,10,11 ans
Poucet : 7,8 ans

Lancement du traitement
cadet



Interprétation Plusieurs choix possibles

```
age <- 16
si age > 20 alors ←
    // traitement senior
finsi
si age > 17 alors ←
    // traitement junior
finsi
si age > 15 alors ←
    // traitement cadet
finsi
si age > 13 alors ←
    // traitement minime
finsi
...
```

Est-ce que l'âge est supérieur à 20 ?

NON

Est-ce que l'âge est supérieur à 17 ?

NON

Est-ce que l'âge est supérieur à 15 ?

OUI

Est-ce que l'âge est supérieur à 13 ?

Rappel

Senior : 21 ans et +
Junior : 18,19,20 ans
Cadet : 16,17 ans
Minime : 14,15 ans
Benjamin : 12,13 ans
Poussin : 9,10,11 ans
Poucet : 7,8 ans

Lancement du traitement
cadet



Interprétation Plusieurs choix possibles

```
age <- 16
si age > 20 alors ←
    // traitement senior
finsi
si age > 17 alors ←
    // traitement junior
finsi
si age > 15 alors ←
    // traitement cadet
finsi
si age > 13 alors ←
    // traitement minime
finsi
...
```

Est-ce que l'âge est supérieur à 20 ? **NON**

Est-ce que l'âge est supérieur à 17 ? **NON**

Est-ce que l'âge est supérieur à 15 ? **OUI**

Est-ce que l'âge est supérieur à 13 ? **OUI**

Rappel

Senior : 21 ans et +
Junior : 18,19,20 ans
Cadet : 16,17 ans
Minime : 14,15 ans
Benjamin : 12,13 ans
Poussin : 9,10,11 ans
Poucet : 7,8 ans

Lancement du traitement
cadet

Lancement du traitement
minime



Interprétation Plusieurs choix possibles

```
age <- 16
si age > 20 alors ←
    // traitement senior
finsi
si age > 17 alors ←
    // traitement junior
finsi
si age > 15 alors ←
    // traitement cadet
finsi
si age > 13 alors ←
    // traitement minime
finsi
...
```

Est-ce que l'âge est supérieur à 20 ? **NON**

Est-ce que l'âge est supérieur à 17 ? **NON**

Est-ce que l'âge est supérieur à 15 ? **OUI**

Est-ce que l'âge est supérieur à 13 ? **OUI**

...

Rappel

Senior : 21 ans et +
Junior : 18,19,20 ans
Cadet : 16,17 ans
Minime : 14,15 ans
Benjamin : 12,13 ans
Poussin : 9,10,11 ans
Poucet : 7,8 ans

Lancement du traitement
cadet

Lancement du traitement
minime



Un seul choix possible

```
si expression alors  
    instructions  
sinon si expression  
alors  
    instructions  
sinon si expression  
alors  
    instructions  
finsi
```

Plusieurs choix possibles

```
si expression alors  
    instructions  
finsi  
si expression alors  
    instructions  
finsi  
si expression alors  
    instructions  
finsi
```



Trois possibilités pour les conditions :

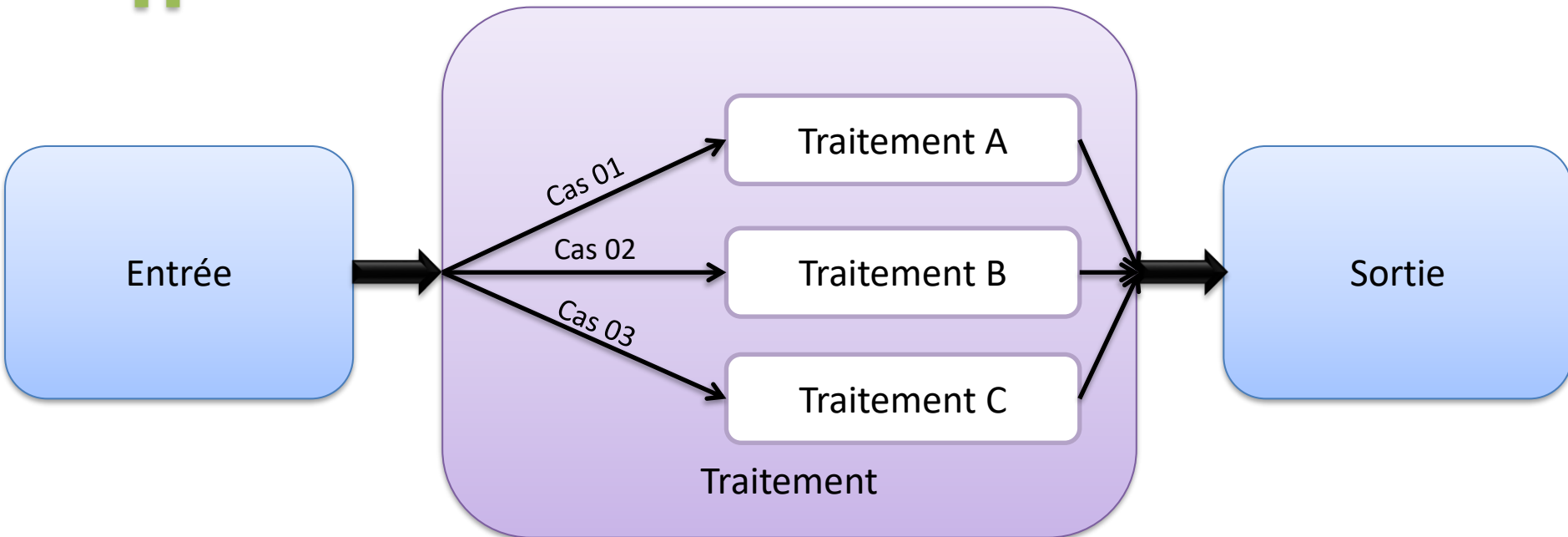
- si [expression] alors... finsi
- si [expression] alors... sinon si [expression] alors... finsi
- si [expression] alors... sinon ... finsi

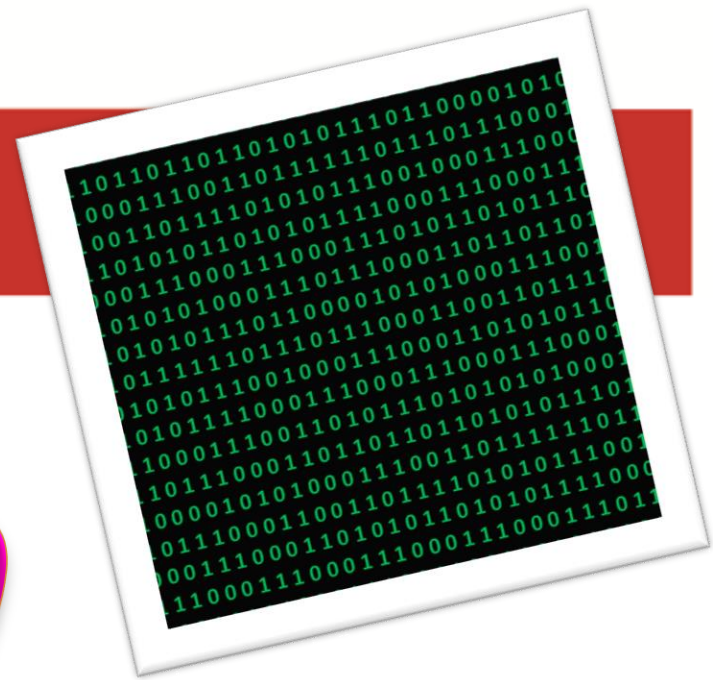
Notez l'absence de « alors » après le « sinon ».

Le « alors » est directement relié au « si ».



Condition : traitement différencié entre plusieurs cas possibles.





0 0 1 0 1 0 0

NOTIONS DE COMPARAISON



On ne peut comparer que
des éléments de même nature.



Nombre entier

Nombre
sans virgule

Nombre réel

Nombre
avec des virgules

Caractère

'a', '!', '3', '\$', '-', ...

Chaîne de caractères

"Bonjour !"
Mots, phrases, ...

Booléen

Vrai ou faux,
Oui ou Non,
1 ou 0



Une variable possède un **nom** et un **type**.

Lors de la déclaration, on indique le type de la variable. La zone mémoire est réservée pour ce type **uniquement**.

Exemples de déclaration :

```
var numeroMois : entier  
var salaire : reel  
var message : chaine  
var a_reussi : booleen
```



Problème pas si trivial !

Rappel du postulat :

On ne peut comparer que des éléments de même nature.

Donc l'ordinateur ne peut pas comparer un nombre entier avec un nombre à virgule.



Problème pas si trivial !

Pourquoi l'ordinateur ne peut pas comparer un nombre entier avec un nombre à virgule ?



Problème pas si trivial !

Pourquoi l'ordinateur ne peut pas comparer un nombre entier avec un nombre à virgule ?

Pour répondre à cette question, il faut reprendre depuis le début



Ca vous parle ?



Livebox Play Fibre



internet jusqu'à 20 méga



appels illimités vers les mobiles ⁽¹⁾
de France métropolitaine



appels illimités vers les fixes ⁽¹⁾
de France métropolitaine et de + de 100
destinations



160 chaînes TV incluses ⁽³⁾
dont 26 en HD ⁽⁴⁾ et 1 en 3D ⁽⁵⁾



enregistreur TV 80Go ⁽⁶⁾
extensible gratuitement sur demande à
240Go

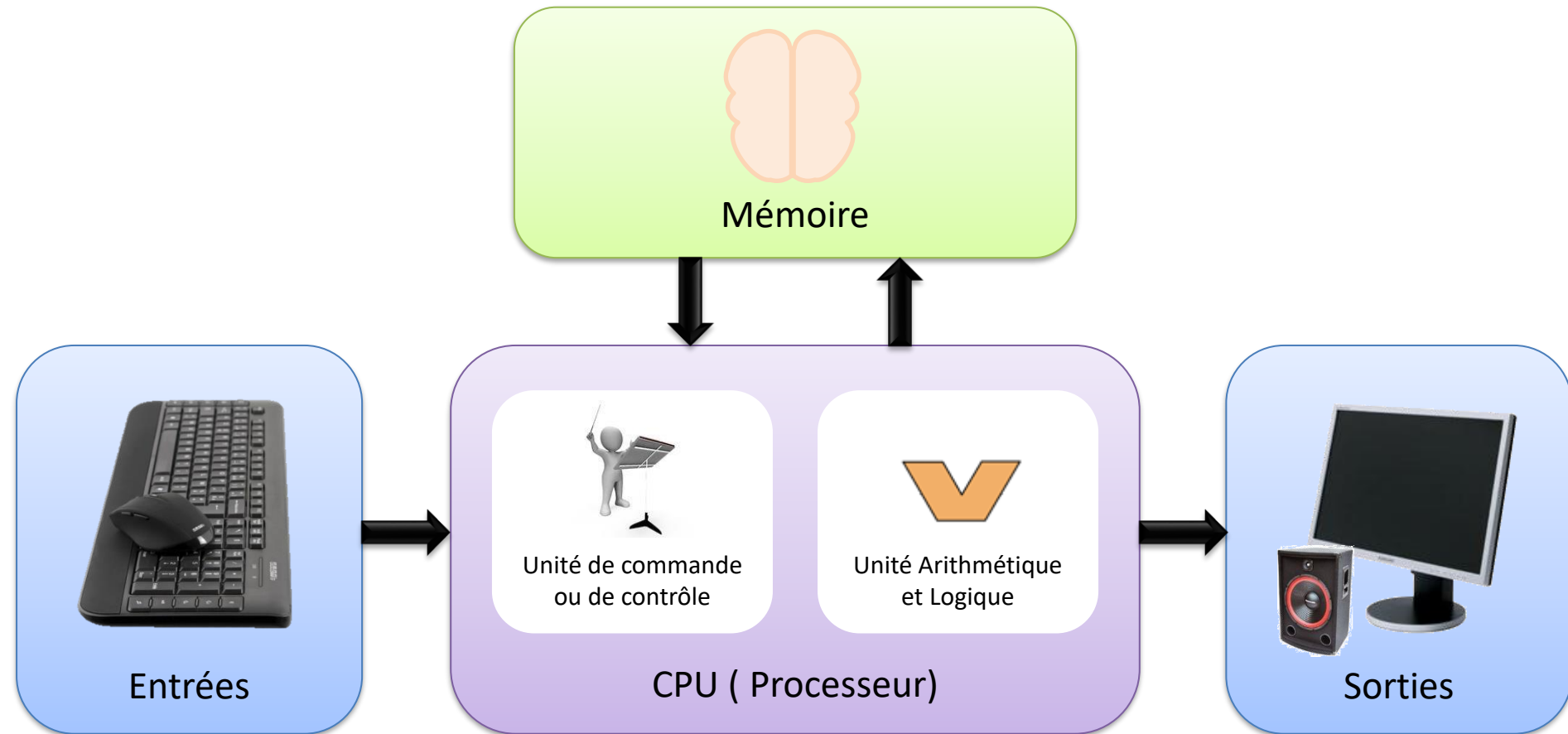


le cloud d'Orange 100Go ⁽⁶⁾



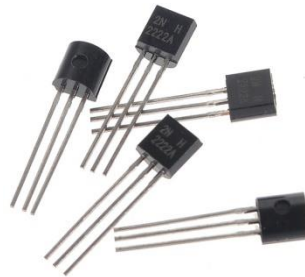


Architecture Von Neumann



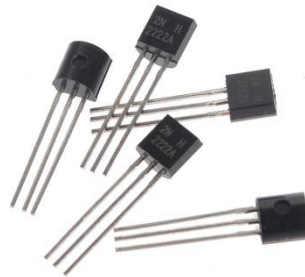


- Le CPU est composé de transistors qui ne peuvent prendre que deux valeurs : l'électricité passe ou l'électricité ne passe pas.





- Le CPU est composé de transistors qui ne peuvent prendre que deux valeurs : l'électricité passe ou l'électricité ne passe pas.



Ca, c'est le début de l'histoire !



C'est quoi le rapport entre nos conditions et des transistors ?

(Nous on aimerait juste apprendre à faire des algorithmes.)



Commençons par une analogie :

- En langage humain, nous avons des lettres.
- Les lettres forment des mots.
- Les mots forment des phrases.
- Les phrases forment des paragraphes.
- Les paragraphes...
- A partir de 26 lettres, nous pouvons créer une infinité de mots.



Continuons avec une autre analogie :

- En mathématiques, nous avons des chiffres.
- Les chiffres forment des nombres.
- Chaque chiffre peut prendre 10 valeurs.
- A partir de là nous pouvons créer une infinité de nombres.



Base 10 vs base 2



Base 10 vs base 2

En mathématiques, nous avons des **chiffres**.

En informatique, nous avons des **bits**.



Base 10 vs base 2

En mathématiques, nous avons des **chiffres**.

Les **chiffres** forment des **nombre**s.

En informatique, nous avons des **bits**.

Les **bits** forment des **octets**.



Base 10 vs base 2

En mathématiques, nous avons des **chiffres**.

Les **chiffres** forment des **nombres**.

Chaque **chiffre** peut prendre **10** valeurs.

En informatique, nous avons des **bits**.

Les **bits** forment des **octets**.

Chaque **bit** peut prendre **2** valeurs.



Base 10 vs base 2

En mathématiques, nous avons des **chiffres**.

Les **chiffres** forment des **nombres**.

Chaque **chiffre** peut prendre **10** valeurs.

A partir de là nous pouvons créer une infinité de **nombres**.

En informatique, nous avons des **bits**.

Les **bits** forment des **octets**.

Chaque **bit** peut prendre **2** valeurs.

A partir de là nous pouvons créer une infinité d'**octets**.



En mathématiques, nous avons des **chiffres**.

Les **chiffres** forment des **nombres**.

Chaque **chiffre** peut prendre **10** valeurs.

A partir de là nous pouvons créer une infinité de **nombres**.

Ca, c'est le système décimal

En informatique, nous avons des **bits**.

Les **bits** forment des **octets**.

Chaque **bit** peut prendre **2** valeurs.

A partir de là nous pouvons créer une infinité d'**octets**.

Ca, c'est le système binaire



1	0	1	0	0	0	1	1
---	---	---	---	---	---	---	---

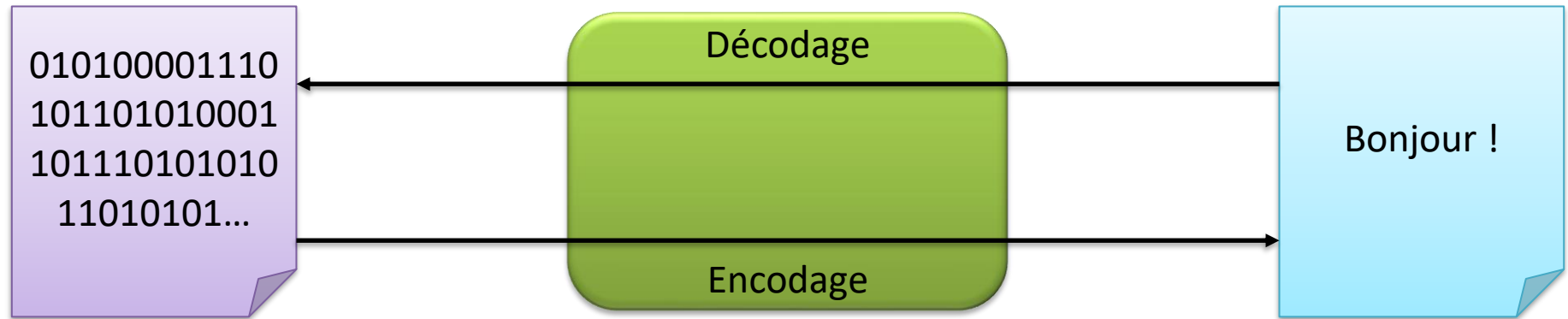


1	0	1	0	0	0	1	1
---	---	---	---	---	---	---	---

Un octet = 8 bits = 1 Byte



Tout est
une histoire de codage



A partir de là nous pouvons définir un codage et faire une traduction entre le système binaire et le langage humain.

Par exemple :

Octet	00000000	00000001	00000010	00000011	00000100
Codage X	a	b	c	d	e



1	0	1	0	0	0	1	1
---	---	---	---	---	---	---	---

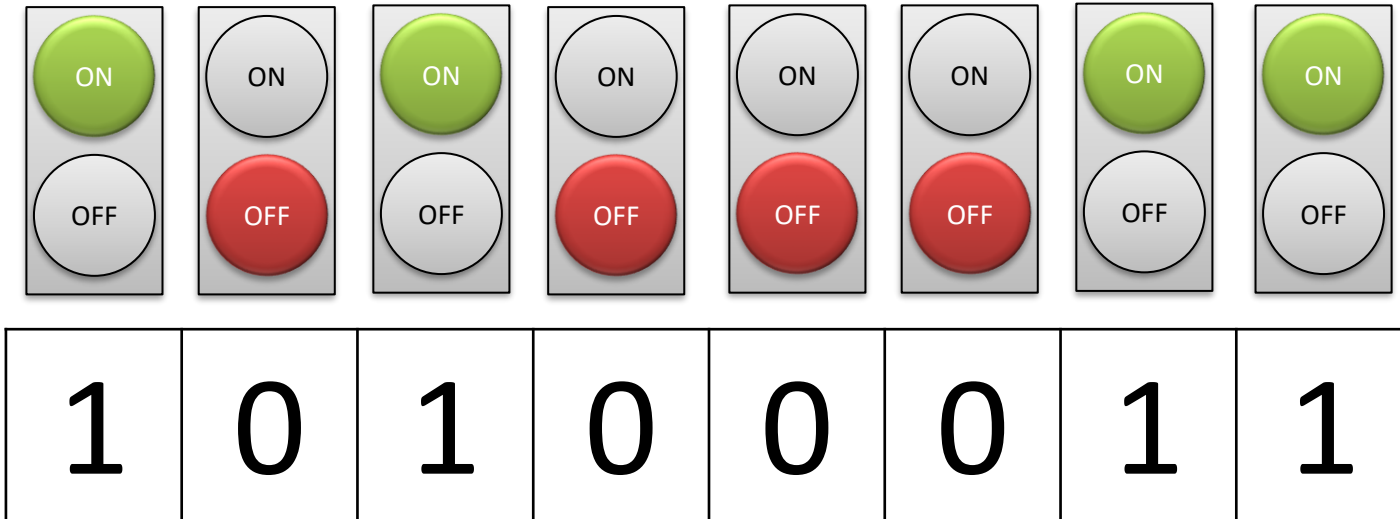


1	0	1	0	0	0	1	1
---	---	---	---	---	---	---	---

Un octet = 8 bits = 1 Byte



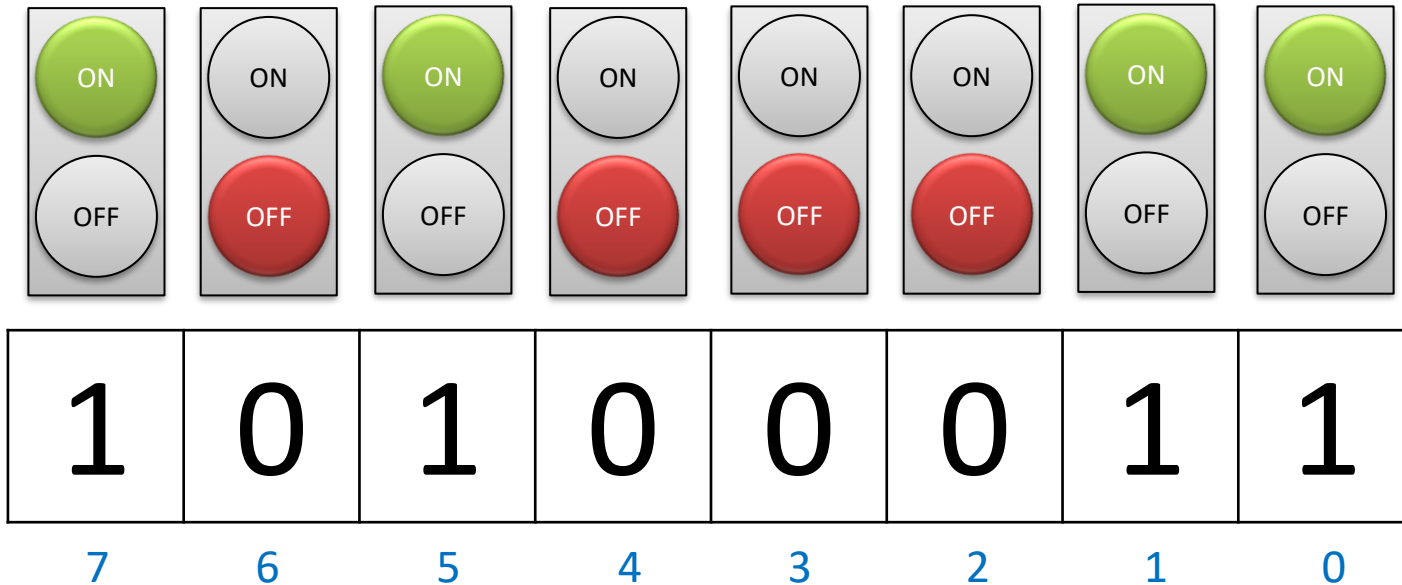
Octet et bit



Un octet = 8 bits = 1 Byte



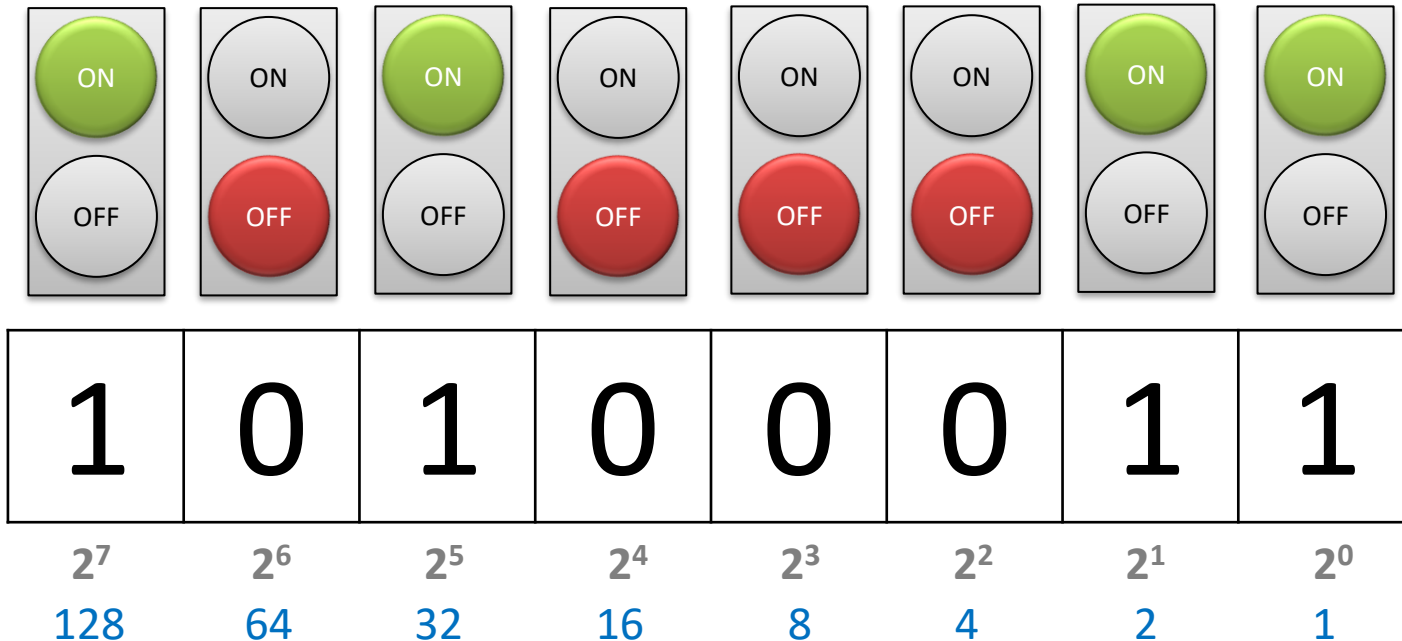
Position du bit



Chaque bit possède une position dans l'octet



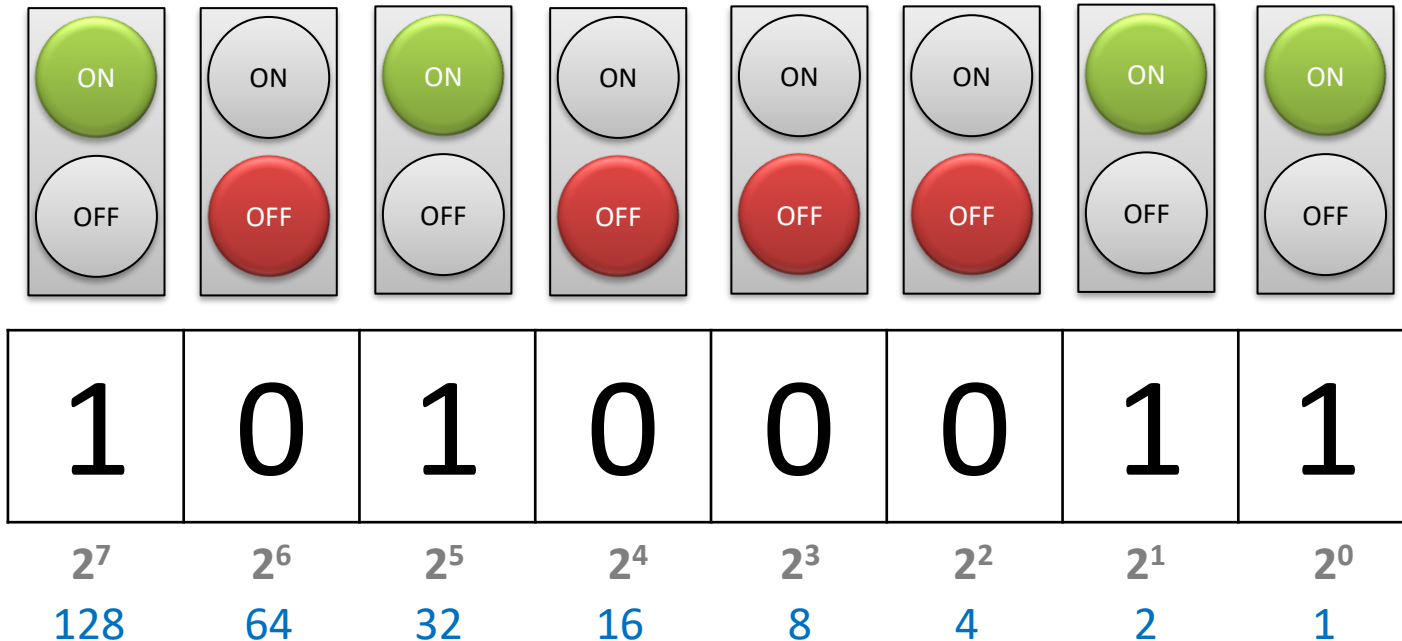
Position du bit



Chaque bit possède une valeur en fonction de sa position dans l'octet



valeur de l'octet



On calcule la valeur de l'octet en
additionnant la valeur des bits à 1



valeur de l'octet

1	0	1	0	0	0	1	1
128	64	32	16	8	4	2	1

$$1 + 2 + 32 + 128 = 163$$



valeur de l'octet

0	0	0	1	0	1	0	0
128	64	32	16	8	4	2	1

A vous !



Et alors ?

Mais, c'est quoi le lien entre tout ça et la question de départ ?

(Pourquoi l'ordinateur ne peut pas comparer un nombre entier avec un nombre à virgule ?)



Type de variable	Taille
Nombre entier	1 octet (de 0 à 255) 2 octets (de 0 à 65 535 ou de -32 768 à 32 767) 4 octets (de 0 à 4 294 967 296 ou de - 2 147 486 648 à 2 147 483 647) 8 octets (de -2^{63} à $+2^{63} - 1$)
Réels	Simple : 4 octets (de $-1.4 * 10^{-45}$ à $+3.4 * 10^{38}$) Double : 8 octets ($4.9 * 10^{-324}$ à $+1.7 * 10^{308}$)
Caractère	2 octets
Chaîne de caractère	Nombre de caractères x 2 octets
Booléen	1 octet

Attention, il peut y avoir des variations en fonction du langage de programmation.



Est-ce similaire ?

0	0	0	1	0	1	0	0
---	---	---	---	---	---	---	---

0	0	0	1	1	0	0	0
---	---	---	---	---	---	---	---



Est-ce similaire ?

0	0	0	1	0	1	0	0
---	---	---	---	---	---	---	---

0	0	0	1	1	0	0	0
---	---	---	---	---	---	---	---

NON



Est-ce similaire ?

1	0	1	0	0	0	1	0
---	---	---	---	---	---	---	---

1	0	1	0	0	0	1	0
---	---	---	---	---	---	---	---



Est-ce similaire ?

1	0	1	0	0	0	1	0
---	---	---	---	---	---	---	---

1	0	1	0	0	0	1	0
---	---	---	---	---	---	---	---

OUI



Est-ce similaire ?

0	0	0	1	0	1	0	0	1	1	0	1	0	0	1	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

1	1	0	1	1	1	0	1	0	0	0	1	1	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---



Est-ce similaire ?

0	0	0	1	0	1	0	0	1	1	0	1	0	0	1	0	1	1	1	1	1	1	0	0	0	0	0	0	0	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

1	1	0	1	1	1	0	1	0	0	0	1	1	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

NON



Comparaison

0	0	0	1	0	1	0	0	1	1	0	1	0	0	1	0	1	1	1	1	1	1	0	0	0	0	0	0	0	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Nombre réel

1	1	0	1	1	1	0	1	0	0	0	1	1	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Nombre entier

Voilà, vous avez la réponse 😊 !



Comparaison

0	0	0	1	0	1	0	0	1	1	0	1	0	0	1	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Nombre réel

1	1	0	1	1	1	0	1	0	0	0	1	1	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Nombre entier

On ne peut comparer que ce qui est comparable

Voilà, vous avez la réponse 😊 !



En vrai la mémoire contient des cases qui contiennent soit 0, soit 1.

0	1	1	1	0	0	1	0	1	0	1	1	1	1	1	0	0	0	1	1	1	0	1	0	0	1	1	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---



En vrai la mémoire contient des cases qui contiennent soit 0, soit 1.

0	1	1	1	0	0	1	0	1	0	1	1	1	1	1	0	0	0	1	1	1	0	1	0	0	1	1	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Si j'écris : `var a : entier`



En vrai la mémoire contient des cases qui contiennent soit 0, soit 1.

0	1	1	1	0	0	1	0	1	0	1	1	1	1	1	0	0	0	1	1	1	0	1	0	0	1	1	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Si j'écris : `var a : entier`

0	1	1	1	0	0	1	0	1	0	1	1	1	1	1	0	0	0	1	1	1	0	1	0	0	1	1	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

a



En vrai la mémoire contient des cases qui contiennent soit 0, soit 1.

0	1	1	1	0	0	1	0	1	0	1	1	1	1	1	0	0	0	1	1	1	0	1	0	0	1	1	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Si j'écris : `var a : entier`

0	1	1	1	0	0	1	0	1	0	1	1	1	1	1	0	0	0	1	1	1	0	1	0	0	1	1	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

a

Si j'écris : `var succes : booléen`



En vrai la mémoire contient des cases qui contiennent soit 0, soit 1.

0	1	1	1	0	0	1	0	1	0	1	1	1	1	1	0	0	0	1	1	1	0	1	0	0	1	1	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Si j'écris : `var a : entier`

0	1	1	1	0	0	1	0	1	0	1	1	1	1	1	0	0	0	1	1	1	0	1	0	0	1	1	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

a

Si j'écris : `var succes : booléen`

0	1	1	1	0	0	1	0	1	0	1	1	1	1	1	0	0	0	1	1	1	0	1	0	0	0	1	1	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

a

succes



Le **type** d'une variable indique à l'ordinateur comment stocker, coder et décoder l'information en mémoire.

Cela signifie :

- Combien de cases mémoire occupe la donnée ?
- Comment « traduire » et utiliser l'ensemble de ces cases ?



Mémoire de départ :

0	1	1	1	0	0	1	0	1	0	1	1	1	1	1	0	0	0	1	1	1	0	1	0	0	1	1	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---



Mémoire de départ :

0	1	1	1	0	0	1	0	1	0	1	1	1	1	1	0	0	0	1	1	1	0	1	0	0	1	1	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Déclaration : `var a : entier`



Mémoire de départ :

0	1	1	1	0	0	1	0	1	0	1	1	1	1	1	0	0	0	1	1	1	0	1	0	0	1	1	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Déclaration : `var a : entier`

0	1	1	1	0	0	1	0	1	0	1	1	1	1	1	0	0	0	1	1	1	0	1	0	0	1	1	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

a



Mémoire de départ :

0	1	1	1	0	0	1	0	1	0	1	1	1	1	1	0	0	0	1	1	1	0	1	0	0	1	1	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Déclaration : `var a : entier`

0	1	1	1	0	0	1	0	1	0	1	1	1	1	1	0	0	0	1	1	1	0	1	0	0	1	1	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Initialisation : `a ← 0`



Mémoire de départ :

0	1	1	1	0	0	1	0	1	0	1	1	1	1	1	0	0	0	1	1	1	0	1	0	0	1	1	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Déclaration : `var a : entier`

0	1	1	1	0	0	1	0	1	0	1	1	1	1	1	0	0	0	1	1	1	0	1	0	0	1	1	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

^aInitialisation : `a ← 0`

0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	1	0	0	0	1	1	1	0	1	0	0	1	1	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

^a



Mémoire de départ :

0	1	1	1	0	0	1	0	1	0	1	1	1	1	1	0	0	0	1	1	1	0	1	0	0	1	1	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

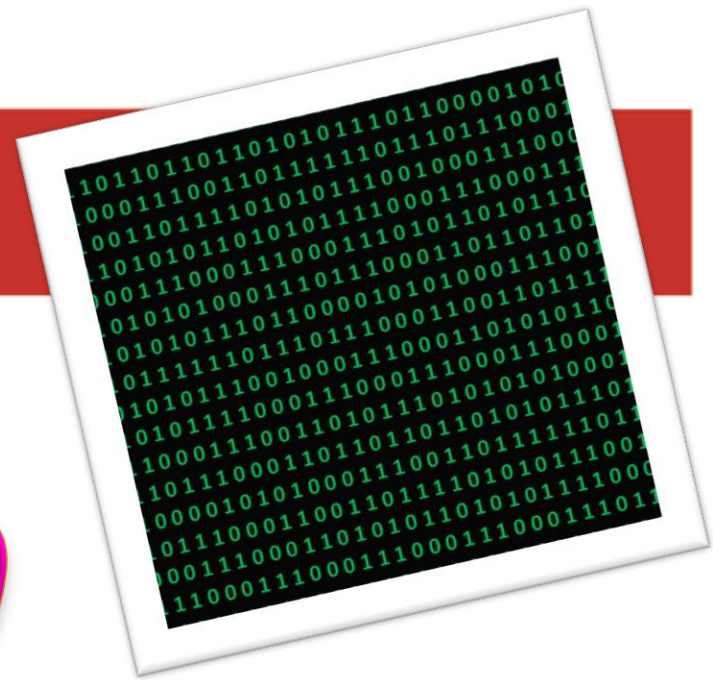
Déclaration : `var a : entier`

0	1	1	1	0	0	1	0	1	0	1	1	1	1	1	0	0	0	1	1	1	0	1	0	0	1	1	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Initialisation : `a ← 0`

0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	1	0	0	0	1	1	1	0	1	0	0	1	1	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Suite de l'algorithme...



0 0 1 0 1 0 0

LES OPÉRATEURS DE COMPARAISON



Petit aparté - La concaténation est une opération entre deux chaînes de caractères.

Par exemple :

« Bon » + « jour »

**Le résultat de cette opération
est « Bonjour ».**



Rappel exemple 3

debut

```
si age > 20 alors
    ecrire « C'est un sénior »
sinon si age > 17 alors
    ecrire « C'est un junior »
sinon si age > 15 alors
    ecrire « C'est un cadet »
sinon si age > 13 alors
    ecrire « C'est un minime »
sinon si age > 11 alors
    ecrire « C'est un benjamin »
finsi
fin
```

**Souvenez-vous de notre
exemple de départ.
Dans les conditions, les
expressions sont des
comparaisons.**



```
debut
```

```
    si age > 20 alors
```

```
        écrire « C'est un sénior »
```

```
    sinon si age > 17 alors
```

```
        écrire « C'est un junior »
```

```
    sinon si age > 15 alors
```

```
        écrire « C'est un cadet »
```

```
    sinon si age > 13 alors
```

```
        écrire « C'est un minime »
```

```
    sinon si age > 11 alors
```

```
        écrire « C'est un benjamin »
```

```
    finsi
```

```
fin
```

**Pour faire des comparaisons,
on utilise des opérateurs de
comparaison.**



Il existe deux types de comparaison :

- Comparaison de type mathématique
- Comparaison logique (basée sur les bits)

Le résultat d'une comparaison est toujours un booléen.



$$a < b$$

$$a \geq b$$

$$a \leq b$$

$$a \neq b$$

$$a > b$$

$$a = b$$



« $a < b$ » se lit « est-ce que a est inférieur à b ? »
« $a = b$ » se lit « est-ce que a est égal à b ? »
« $a \neq b$ » se lit « est-ce que a est différent de b »?
...

**La réponse a chacune de ces questions est
« vrai » ou « faux ». C'est un booléen.**



a ET b

a OU b

NON a

N'oubliez pas: toutes ces expressions renvoient un booléen



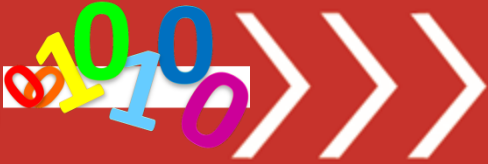
Est-ce que les
deux
conditions sont
satisfaites en
même temps ?

a ET b

a OU b

NON a

N'oubliez pas: toutes ces expressions renvoient un booléen



Est-ce que les deux conditions sont satisfaites en même temps ?

a ET b

Est-ce qu'au moins une des deux conditions est satisfaite ?

a OU b

NON a

N'oubliez pas: toutes ces expressions renvoient un booléen



Est-ce que les deux conditions sont satisfaites en même temps ?

a ET b

Est-ce qu'au moins une des deux conditions est satisfaite ?

a OU b

Si **a** est vrai, alors **NON a** vaut faux. Et inversement.

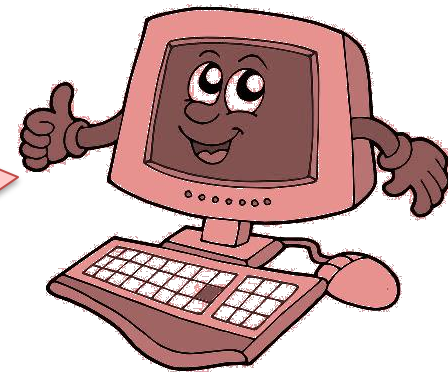
NON a

N'oubliez pas: toutes ces expressions renvoient un booléen



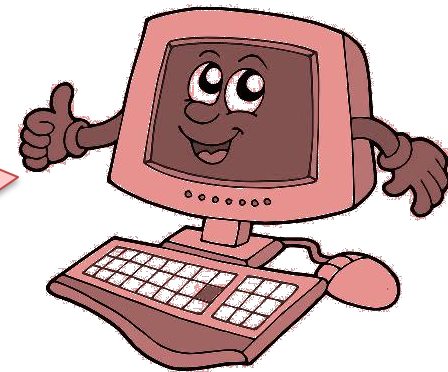
N'oubliez pas :

- Avec les opérateurs arithmétiques, on compare des nombres : $a < b$, $a = b$, $a > b$, etc.
- Avec les opérateurs logiques, on compare des booléens : a et b , a ou b , non a .





N'oubliez pas :
Le résultat d'une comparaison
entre deux variables de type T
est un booléen.





Attention !!!!!

si expression alors



si **booléen** alors

$a < b \Rightarrow$ résultat booléen

si $a < b$ alors

$a \leftarrow 2 \Rightarrow$ résultat « 2 est mis dans une case »

~~si $a \leftarrow 2$ alors~~

$a + 3 \Rightarrow$ résultat entier

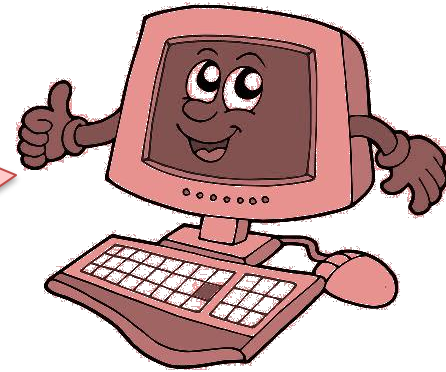
~~si $a + 3$ alors~~



Booléen

N'oubliez pas :

- Un booléen permet de répondre à la question « est-ce que ... ? »
- Il peut posséder deux valeurs : vrai ou faux.



Exemple : Est-ce que 3 est supérieur à 5?





Entier  Entier  Entier



Entier Entier \rightarrow Entier

Entier Entier \rightarrow Booleen



Entier Entier \rightarrow Entier

Entier Entier \rightarrow Booleen

Booleen Booleen \rightarrow Booleen

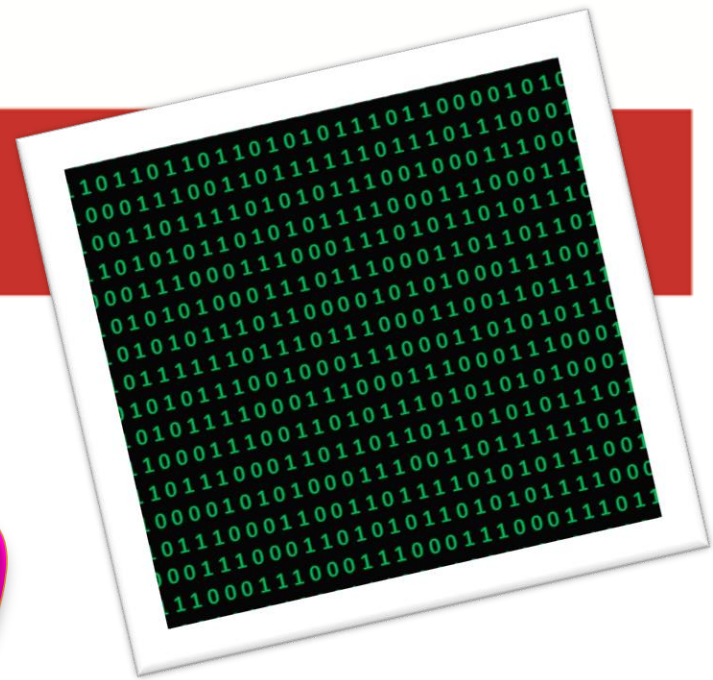


```
var age : entier
age <- 16
si age < 7 ou age > 60 alors
    // traitement « pas de sport »
sinon si age >= 7 et age < 9 alors
    // traitement poucet
sinon si age > 8 et age <= 11 alors
    // traitement poussin
sinon si age > 11 et age < 14 alors
    // traitement benjamin
sinon si age >= 14 et age <= 15 alors
    // traitement minime
sinon si age > 15 et age <= 17 alors
    // traitement cadet
sinon si age >= 18 et age < 21 alors
    // traitement junior
sinon
    // traitement senior
finsi
```

**L'expression d'une
condition doit être un
booléen.**

=

**Le résultat de
l'opération doit être un
booléen.**



0 0 1 0 1 0 0

Notion de vocabulaire

BLOC ET IMBRICATION



```
debut
  var somme : entier
  lire somme //valeur venant du clavier
  si somme ≠ 0 alors
    si somme > 0 alors
      ecrire « Yes, j'ai de l'argent ! »
    sinon si somme < 0 alors
      ecrire « Oh non, je dois de l'argent... »
    finsi
  finsi
fin
```

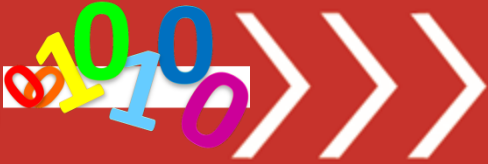


Bloc d'instructions



debut

fin



debut



fin

Bloc d'instructions



debut

```
var somme : entier  
somme <- 0
```

fin

Bloc d'instructions



debut

```
var somme : entier  
somme <- 0  
si somme ≠ 0 alors
```

```
finsi
```

fin

Bloc d'instructions



debut

```
var somme : entier  
somme <- 0  
si somme ≠ 0 alors
```



```
finsi
```

fin

Bloc d'instructions

Bloc d'instructions



debut

```
var somme : entier  
somme <- 0  
si somme ≠ 0 alors  
    si somme > 0 alors  
  
    sinon  
  
finsi  
finsi
```

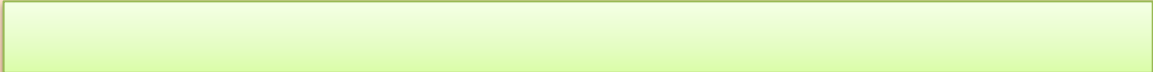
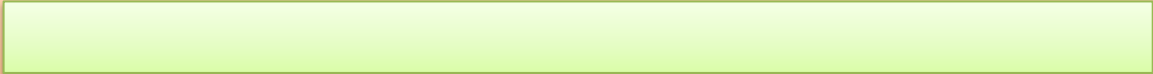
fin

Bloc d'instructions

Bloc d'instructions



debut

```
var somme : entier  
somme <- 0  
si somme ≠ 0 alors  
  si somme > 0 alors  
      
  sinon  
      
  finsi  
finsi
```

fin

Bloc d'instructions

Bloc d'instructions

Bloc d'instructions



debut

```
var somme : entier  
somme <- 0  
si somme ≠ 0 alors
```

```
  si somme > 0 alors
```

```
    écrire « Yes, j'ai de l'argent ! »
```

```
  sinon
```

```
    écrire « Oh non, je dois de l'argent... »
```

```
  finsi
```

```
finsi
```

fin

Bloc d'instructions

Bloc d'instructions

Bloc d'instructions



```
debut
```

```
  var somme : entier
```

```
  somme <- 0
```

```
  si somme ≠ 0 alors
```

```
    si somme > 0 alors
```

```
      écrire « Yes, j'ai de l'argent ! »
```

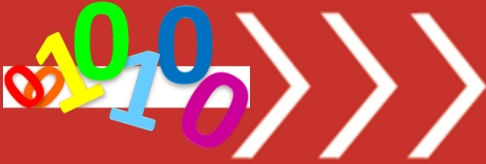
```
    sinon
```

```
      écrire « Oh non, je dois de l'argent... »
```

```
    finsi
```

```
  finsi
```

```
fin
```



```
debut
```

```
  var somme : entier
```

```
  somme <- 0
```

```
  si somme ≠ 0 alors
```

```
    si somme > 0 alors
```

```
      écrire « Yes, j'ai de l'argent ! »
```

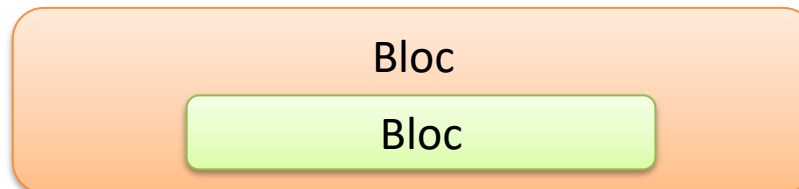
```
    sinon
```

```
      écrire « Oh non, je dois de l'argent... »
```

```
    finsi
```

```
  finsi
```

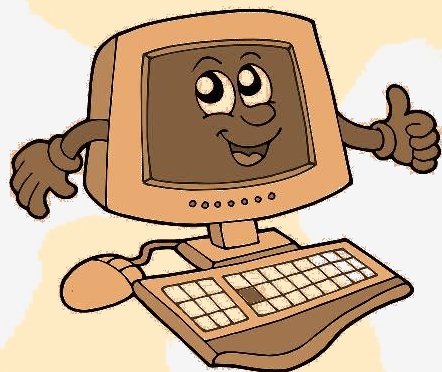
```
fin
```





INVENTAIRE

AVATAR



NOM : *Algorithmique*

SAC A DOS



E/S

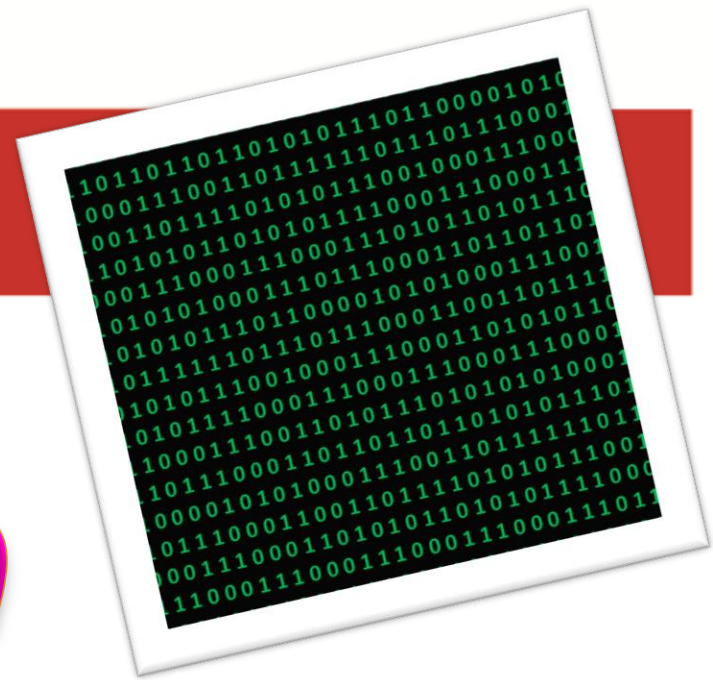


SI

< ≤ =

≥ > ≠

ET OU
NON



0 0 1 0 1 0 0

Exercices

UN PEU D'ENTRAÎNEMENT



Règles de calcul des impôts



- Si la personne est au SMIC, alors prélever 10% de son salaire.
- Si la personne est de classe moyenne, prélever 25% de son salaire.
- Si la personne est riche, ne surtout pas toucher son argent.



Ecrire un algorithme qui calcule un impôt en fonction des règles de calcul suivantes :

- Si la personne gagne moins de 30000 euros par an, alors l'impôt est égal à 10% du salaire.
- Si la personne gagne entre 30000 et 60000 euros par an, l'impôt est égal à 25% de son salaire.
- Si la personne gagne plus de 60000 euros par an, il n'y a pas d'impôt.





debut

//Déclarations

var salaire : reel

var impots : reel

//Initialisations

lire salaire

impots <- 0

//Algo

si salaire < 30000.0 alors

 impots <- salaire * 0.1

sinon si salaire >= 30000.0 et salaire <= 60000.0 alors

 impots <- salaire * 0.25

finsi

fin



Ecrire un algorithme qui vérifie si une année est bissextile.

Attention, une année est bissextile :

- Si l'année est divisible par 4 et non divisible par 100
- Ou si l'année est divisible par 400
- Sinon, elle n'est pas bissextile.