

# Représentation des données: types et valeurs de base

Lionel Avon, Ronan Charpentier, Anne Dominguez, Jean-Luc  
Leloire, Olivier Longuet

2 juillet 2019



# Le programme

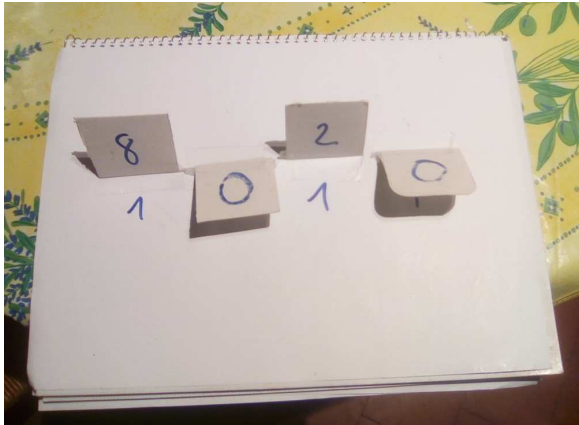
Contenus	Capacités attendues	Commentaires
Écriture d'un entier positif dans une base $b \geq 2$	Passer de la représentation d'une base dans une autre.	Les bases 2, 10 et 16 sont privilégiées.

# Se mesurer en binaire



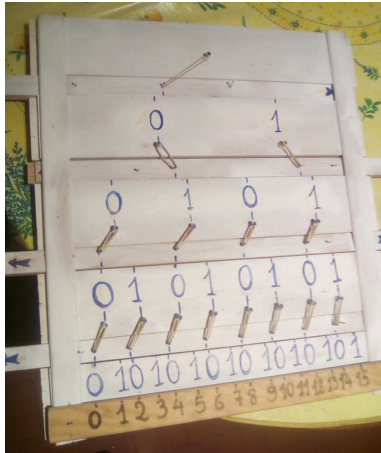
$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$

# Cartes binaires



Site de Lionel Avon

# Un arbre binaire




# Le programme

Contenus	Capacités attendues	Commentaires
Représentation binaire d'un entier relatif	Évaluer le nombre de bits nécessaires à l'écriture en base 2 d'un entier, de la somme ou du produit de deux nombres entiers. Utiliser le complément à 2.	Il s'agit de décrire les tailles courantes des entiers (8, 16, 32 ou 64 bits). Il est possible d'évoquer la représentation des entiers de taille arbitraire de Python.

Contenus	Capacités attendues	Commentaires
Représentation approximative des nombres réels : notion de nombre flottant	Calculer sur quelques exemples la représentation de nombres réels : 0.1, 0.25 ou $1/3$ . $0.2 + 0.1$ n'est pas égal à 0.3.	Il faut éviter de tester l'égalité de deux flottants. Aucune connaissance précise de la norme IEEE-754 n'est exigible.



# La norme IEEE-754

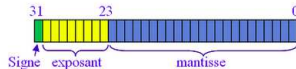
	Signe	Exposant	Mantisse
Valeur	+	$2^{-126}$ (dénormalisé)	0 (dénormalisé)
Encodé :	0	0	0
Binaire			
Valeur		<input type="text"/>	
Binaire	0b	<input type="text"/>	
Hexadécimal	0x	<input type="text"/>	
Erreur de conversion			

Un nombre décimal en binaire sera du type : 1011.1101

qui signifie :  $1.2^3 + 0.2^2 + 1.2^1 + 1.2^0 + 1.2^{-1} + 1.2^{-2} + 0.2^{-3} + 1.2^{-4} = 8 + 0 + 2 + 1 + 0,5 + 0,25 + 0,0625 = 11,8125$  en décimal.

## - Codage d'un réel en simple précision sur 32 bits.

Un nombre float simple précision est stocké dans un mot de 32 bits :



Le signe est codé sur 1 bit, l'exposant sur 8 bits et la mantisse sur 23 bits

L'exposant est décalé de  $2^8 - 1 = 127$  et donc ira de -126 à 127.

Site de Jean-Luc Leloire

TD

# Convertir un décimal un binaire

## Convertir un décimal positif en binaire : format ... , ...

Saisir un nombre décimal :

11.8125

Convertir

( Saisir un . pour la virgule ou saisir une fraction. )

Une écriture en base 2 :

( Partie fractionnaire sur 2 octets au maximum. )

## Convertir un décimal positif en binaire : format ... , ...

Saisir un nombre décimal :

11.8125

Convertir

( Saisir un . pour la virgule ou saisir une fraction. )

Une écriture en base 2 :

1011,1101

( Partie fractionnaire sur 2 octets au maximum. )

Autre format :  $1,0111101 \times 2^3$

# Le programme

Contenus	Capacités attendues	Commentaires
Valeurs booléennes : 0, 1. Opérateurs booléens : and, or, not. Expressions booléennes	Dresser la table d'une expression booléenne.	Le ou exclusif (xor) est évoqué. Quelques applications directes comme l'addition binaire sont présentées. L'attention des élèves est attirée sur le caractère séquentiel de certains opérateurs booléens.

# Un tour de magie

Choisir un nombre  
entre 1 et 30.

1, 3, 5, 7, 9, 11, 13,  
15, 17, 19, 21, 23,  
25, 27, 29

2, 3, 6, 7, 10, 11,  
14, 15, 18, 19, 22,  
23, 26, 27, 30

4, 5, 6, 7, 12, 13,  
14, 15, 20, 21, 22,  
23, 28, 29, 30

8, 9, 10, 11, 12, 13,  
14, 15, 24, 25, 26,  
27, 28, 29, 30

16, 17, 18, 19, 20,  
21, 22, 23, 24, 25,  
26, 27, 28, 29, 30

# Logique booléenne

a	b	not a	a or b	a and b	a xor b
False	False				
False	True				
True	False				
True	True				

# Logique booléenne

a	b	not a	a or b	a and b	a xor b
False	False	True			
False	True	True			
True	False	False			
True	True	False			



# Logique booléenne

a	b	not a	a or b	a and b	a xor b
False	False	True	False		
False	True	True			
True	False	False			
True	True	False			

# Logique booléenne

a	b	not a	a or b	a and b	a xor b
False	False	True	False		
False	True	True	True		
True	False	False	True		
True	True	False	True		

# Logique booléenne

a	b	not a	a or b	a and b	a xor b
False	False	True	False		
False	True	True	True		
True	False	False	True		
True	True	False	True	True	

# Logique booléenne

a	b	not a	a or b	a and b	a xor b
False	False	True	False	False	
False	True	True	True	False	
True	False	False	True	False	
True	True	False	True	True	

# Logique booléenne

a	b	not a	a or b	a and b	a xor b
False	False	True	False	False	False
False	True	True	True	False	True
True	False	False	True	False	True
True	True	False	True	True	False

NB en Python `a xor b` se note `a != b`

# Bit-à-bit (bitwise)

```
>>> x=60
```

```
>>> y=2
```

```
>>> x^y
```

# Bit-à-bit (bitwise)

```
>>> x=60
```

```
>>> y=2
```

```
>>> x^y
```

```
62
```

# Bit-à-bit (bitwise)

```
>>> x=60
```

```
>>> y=2
```

```
>>> x^y
```

```
62
```

```
>>> x|y, x&y
```



# Bit-à-bit (bitwise)

```
>>> x=60
```

```
>>> y=2
```

```
>>> x^y
```

```
62
```

```
>>> x|y, x&y
```

```
62,0
```

# Bit-à-bit (bitwise)

```
>>> x=60
```

```
>>> y=2
```

```
>>> x^y
```

```
62
```

```
>>> x|y, x&y
```

```
62,0
```

```
>>> bin(x),bin(y)
```

# Bit-à-bit (bitwise)

```
>>> x=60
>>> y=2
>>> x^y

62
>>> x|y, x&y
62,0
>>> bin(x),bin(y)
('0b111100', '0b10')
```

# Bit-à-bit (bitwise)

```
>>> x=60
>>> y=2
>>> x^y

62
>>> x|y, x&y
62,0
>>> bin(x),bin(y)
('0b111100', '0b10')
```

# Bit-à-bit (bitwise)

```
>>> x=60
>>> y=2
>>> x^y

62
>>> x|y, x&y
62,0
>>> bin(x),bin(y)
('0b111100', '0b10')
>>> ~12
-13
```

# Valar Morghulis



# Valar Morghulis



A chaque personnage on associe un vecteur booléen  $(x_1; x_2; \dots; x_n)$ .

# Valar Morghulis



A chaque personnage on associe un vecteur booléen  $(x_1; x_2; \dots; x_n)$ .

Par exemple on note  $x_1$  : le personnage est féminin.



# Valar Morghulis



A chaque personnage on associe un vecteur booléen  $(x_1; x_2; \dots; x_n)$ .

Par exemple on note

$x_1$  : le personnage est féminin.

$x_2$  : le personnage est gentil

# Valar Morghulis



A chaque personnage on associe un vecteur booléen  $(x_1; x_2; \dots; x_n)$ .

Par exemple on note

$x_1$  : le personnage est féminin.

$x_2$  : le personnage est gentil

Déterminer des variables suffisamment précises et suffisamment nombreuses pour distinguer les personnages.

# Une question facile

```
>>> x=True
>>> y=False
>>> z=(x and y) or (not y or x)
>>> t= x or y and not z or not (x or y)
>>> z,t
```

# Une question facile

```
>>> x=True
>>> y=False
>>> z=(x and y) or (not y or x)
>>> t= x or y and not z or not (x or y)
>>> z,t

True, True
```

# Une question plus difficile

Peut-on déterminer les booléens  $x_1, x_2, x_3, x_4, x_5$  de façon à ce que toutes les clauses suivantes soient satisfaites ?

$x_1 \vee x_2 \vee x_3$	$x_1 \vee x_2 \vee \bar{x}_5$	$x_2 \vee x_3 \vee x_4$	$x_1 \vee x_3 \vee \bar{x}_5$
$\bar{x}_1 \vee x_2 \vee x_4$	$\bar{x}_2 \vee x_4 \vee \bar{x}_5$	$\bar{x}_3 \vee x_4 \vee x_5$	$x_2 \vee \bar{x}_3 \vee \bar{x}_4$
$x_1 \vee x_4 \vee x_5$	$x_3 \vee x_4 \vee x_5$	$x_2 \vee x_4 \vee \bar{x}_5$	$\bar{x}_3 \vee \bar{x}_4 \vee \bar{x}_5$

# Une question plus difficile

Peut-on déterminer les booléens  $x_1, x_2, x_3, x_4, x_5$  de façon à ce que toutes les clauses suivantes soient satisfaites ?

$x_1 \vee x_2 \vee x_3$	$x_1 \vee x_2 \vee \bar{x}_5$	$x_2 \vee x_3 \vee x_4$	$x_1 \vee x_3 \vee \bar{x}_5$
$\bar{x}_1 \vee x_2 \vee x_4$	$\bar{x}_2 \vee x_4 \vee \bar{x}_5$	$\bar{x}_3 \vee x_4 \vee x_5$	$x_2 \vee \bar{x}_3 \vee \bar{x}_4$
$x_1 \vee x_4 \vee x_5$	$x_3 \vee x_4 \vee x_5$	$x_2 \vee x_4 \vee \bar{x}_5$	$\bar{x}_3 \vee \bar{x}_4 \vee \bar{x}_5$

Quelle stratégie pour résoudre des problèmes avec des milliers de littéraux et de clauses ?

# Une question plus difficile

Peut-on déterminer les booléens  $x_1, x_2, x_3, x_4, x_5$  de façon à ce que toutes les clauses suivantes soient satisfaites ?

$x_1 \vee x_2 \vee x_3$	$x_1 \vee x_2 \vee \bar{x}_5$	$x_2 \vee x_3 \vee x_4$	$x_1 \vee x_3 \vee \bar{x}_5$
$\bar{x}_1 \vee x_2 \vee x_4$	$\bar{x}_2 \vee x_4 \vee \bar{x}_5$	$\bar{x}_3 \vee x_4 \vee x_5$	$x_2 \vee \bar{x}_3 \vee \bar{x}_4$
$x_1 \vee x_4 \vee x_5$	$x_3 \vee x_4 \vee x_5$	$x_2 \vee x_4 \vee \bar{x}_5$	$\bar{x}_3 \vee \bar{x}_4 \vee \bar{x}_5$

Quelle stratégie pour résoudre des problèmes avec des milliers de littéraux et de clauses ? IRISA

# Le programme

Contenus	Capacités attendues	Commentaires
Représentation d'un texte en machine. Exemples des encodages ASCII, ISO-8859-1, Unicode	Identifier l'intérêt des différents systèmes d'encodage. Convertir un fichier texte dans différents formats d'encodage.	Aucune connaissance précise des normes d'encodage n'est exigible.



# Hum, des donuts !

