FICHE 1 • 5e

Données et réseaux

Les données informatiques

Bits, octets…

Objectif Comprendre comment les données sont représentées et distinguer un bit d'un octet.

Le cours

Un ordinateur manipule des données : textes, images, sons et programmes. Ces données sont représentées par des suites de **0 et de 1** : c'est le **codage binaire**. Un **bit** est un chiffre binaire, égal à 0 ou à 1. Un **octet** est un groupe de **8 bits**.

1	0	1	0	1	1	0	1

8 bits regroupés = 1 octet

Un octet permet de représenter **256 valeurs différentes**. Si on l'utilise pour coder un entier positif ou nul, ses valeurs vont de **0 à 255**. Pour coder davantage de valeurs, on utilise plusieurs octets.

À retenir 1 octet = 8 bits. Le nombre 255 est la plus grande valeur entière non signée codée sur un octet. Un caractère peut occuper un ou plusieurs octets selon le codage utilisé.

Je m'entraîne

1. Complète : un bit vaut ___ ou ___. Un octet contient ___ bits.

2. Entoure les suites qui forment un octet :
1011 11001010 00000000 111111111

3. Un fichier contient 6 octets. Combien contient-il de bits ?
Peut-on coder l'entier 300 sur un seul octet ? Justifie.

4. Une donnée occupe 24 bits. Combien cela représente-t-il d'octets ? Explique ton calcul.

Données et réseaux

Les données informatiques

Du binaire au décimal

La valeur de chaque bit sur 16 bits

Chaque colonne vaut le double de celle située à sa droite. Le bit le plus à droite vaut **1**. Les valeurs vont de **1 à 32 768**.

32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1

← Vers les grandes valeurs Vers les petites valeurs →

Méthode Aligne les bits à partir de la droite, sous les valeurs 1, 2, 4, 8, 16… Additionne uniquement les valeurs des colonnes contenant **1**. Une colonne contenant **0** n'ajoute rien.

Un exemple sur 4 bits

8	4	2	1
1	0	1	1

1011 en binaire
= 8 + 2 + 1 = **11 en décimal**

Je m'entraîne de 4 à 10 bits

Convertis chaque nombre binaire en décimal. Écris le résultat dans la case après chaque nombre. Utilise le schéma en partant de sa droite.

4 bits	**a) 1010**		**b) 1101**	
5 bits	**a) 10011**		**b) 11100**	
6 bits	**a) 101010**		**b) 110101**	
7 bits	**a) 1001101**		**b) 1110010**	
8 bits	**a) 10010110**		**b) 11100011**	
9 bits	**a) 101010101**		**b) 110001011**	
10 bits	**a) 1001100110**		**b) 1110101011**	

Données et réseaux

Les données informatiques

Des multiples…

Objectif Lire une capacité de stockage et convertir des octets, Ko, Mo, Go et To.

Le cours

La **taille d'un fichier** et la **capacité d'un support de stockage** se mesurent en octets. Les multiples permettent d'exprimer de grandes quantités. Dans ces fiches, chaque changement d'unité correspond à un facteur **1 000**.

1 Ko	1 000 octets
1 Mo	1 000 Ko = 1 000 000 octets
1 Go	1 000 Mo = 1 000 000 000 octets
1 To	1 000 Go = 1 000 000 000 000 octets
1 Po	1 000 To = 1 000 000 000 000 000 octets

Pour convertir vers une unité plus petite, on multiplie par 1 000 à chaque étape. Vers une unité plus grande, on divise par 1 000 à chaque étape.

Exemples 3 Mo = 3 × 1 000 = 3 000 Ko.
5 000 Mo = 5 000 ÷ 1 000 = 5 Go.
Vocabulaire Ko : kilooctet • Mo : mégaoctet • Go : gigaoctet
To : téraoctet • Po : pétaoctet

Je m'entraîne

1. Complète : 4 Mo = _____ Ko ; 2 Go = _____ Mo.

2. Convertis 7 000 Mo en Go, puis 6 000 octets en Ko.

3. Une clé dispose de 2 Go libres. Combien de fichiers de 250 Mo peut-elle encore contenir ? On néglige l'espace de gestion du support.

Données et réseaux

Les données informatiques

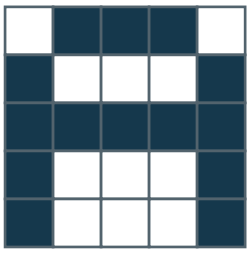
Le codage des images en binaire

Objectif Passer d'une image en noir et blanc à une grille de bits, et inversement.

Le cours

Une image numérique est composée de petits éléments appelés **pixels**. Dans notre modèle d'image en noir et blanc, un pixel est codé par **1 bit**. On choisit la convention : **0 = blanc** et **1 = noir**.

Pour lire ou coder l'image, on commence en haut à gauche, on lit chaque ligne de gauche à droite, puis on passe à la ligne suivante. Il faut connaître le **nombre de colonnes et de lignes** pour reconstruire l'image.



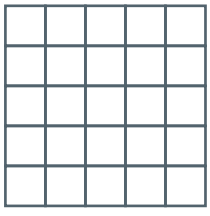
01110
10001
11111
10001
10001

5 colonnes × 5 lignes = 25 pixels
Donc 25 bits pour les pixels de cette image.

À retenir Nombre de pixels = colonnes × lignes. Dans ce modèle, nombre de bits = nombre de pixels. Un vrai fichier contient aussi des informations sur l'image ; la couleur demande un codage plus riche.

Je m'entraîne

1. Colorie la grille avec ce code. Quel symbole reconnais-tu ?
01110 / 10001 / 10000 / 10001 / 01110



Symbole : _____

2. Une image a 8 colonnes et 6 lignes. Calcule son nombre de pixels puis le nombre de bits nécessaires dans notre modèle.

3. Sur une feuille quadrillée, dessine un symbole de 5 × 5 pixels.
Écris ses cinq lignes de code, puis fais-le décoder à un camarade.

Données et réseaux

Les données informatiques

Le codage du texte

Objectif Utiliser une table de correspondance pour coder et décoder un texte.

Le cours

Pour représenter un texte, on associe un **nombre à chaque caractère** : lettre, chiffre, espace ou signe de ponctuation. L'ordinateur représente ensuite ces nombres en binaire.

Le **code ASCII** contient 128 codes, de 0 à 127. Il distingue les majuscules et les minuscules, mais ne code pas les lettres accentuées. **Unicode** représente les caractères de très nombreuses langues.

Exemple A → 65 → 01000001 (écrit sur 8 bits).
Le caractère « 1 » a pour code ASCII 49 : il se distingue de la valeur numérique 1.

La table ASCII complète de 0 à 127

0 **NUL**	1 **SOH**	2 **STX**	3 **ETX**	4 **EOT**	5 **ENQ**	6 **ACK**	7 **BEL**	8 **BS**	9 **HT**	10 **LF**	11 **VT**	12 **FF**	13 **CR**	14 **SO**	15 **SI**
16 **DLE**	17 **DC1**	18 **DC2**	19 **DC3**	20 **DC4**	21 **NAK**	22 **SYN**	23 **ETB**	24 **CAN**	25 **EM**	26 **SUB**	27 **ESC**	28 **FS**	29 **GS**	30 **RS**	31 **US**
32 **ESP**	33 **!**	34 **"**	35 **#**	36 **$**	37 **%**	38 **&**	39 **'**	40 **(**	41 **)**	42 *****	43 **+**	44 **,**	45 **-**	46 **.**	47 **/**
48 **0**	49 **1**	50 **2**	51 **3**	52 **4**	53 **5**	54 **6**	55 **7**	56 **8**	57 **9**	58 **:**	59 **;**	60 **<**	61 **=**	62 **>**	63 **?**
64 **@**	65 **A**	66 **B**	67 **C**	68 **D**	69 **E**	70 **F**	71 **G**	72 **H**	73 **I**	74 **J**	75 **K**	76 **L**	77 **M**	78 **N**	79 **O**
80 **P**	81 **Q**	82 **R**	83 **S**	84 **T**	85 **U**	86 **V**	87 **W**	88 **X**	89 **Y**	90 **Z**	91 **[**	92 ****	93 **]**	94 **^**	95 **_**
96 **`**	97 **a**	98 **b**	99 **c**	100 **d**	101 **e**	102 **f**	103 **g**	104 **h**	105 **i**	106 **j**	107 **k**	108 **l**	109 **m**	110 **n**	111 **o**
112 **p**	113 **q**	114 **r**	115 **s**	116 **t**	117 **u**	118 **v**	119 **w**	120 **x**	121 **y**	122 **z**	123 **{**	124 **|**	125 **}**	126 **~**	127 **DEL**

Dans chaque case : **code décimal en haut**, caractère en bas. **ESP** = espace.
Les codes **0 à 31** et **127** sont des commandes, indiquées par leur abréviation.

Je m'entraîne

1. Décode : 66 114 97 118 111 32 33

2. Écris les codes ASCII de **Salut !** en conservant l'espace.

3. Décode le message du cours initial :
78 111 117 115 32 115 111 109 109 101 115 32 101 110 32 99 111 117 114 115 32 100 101 32 84 101 99 104 110 111 108 111 103 105 101 32 33