

# Cours d'informatique (DF et OC)

Lycée Blaise Cendrars

11 septembre 2024

Ce document est publié sous licences  
GFDL sans sections invariantes et GPL  
dernière version dans les deux cas.  
Le texte de ces licences est disponible à la FSF.

# Évaluations

Désagréables  $\longleftrightarrow$  agréables<sup>1</sup> ?



Description des évaluations

# DISCIPLINE FONDAMENTALE

# Table des matières

- 1 DF Systèmes numériques
  - Problématique
  - Matériel
  - Systèmes d'exploitation
  - Licences
  - Applications
  - Réseau
  - Sécurité

- 1 DF Systèmes numériques
- 2 DF Informations et données
- 3 DF Algo. et programmation
- 4 DF Informatique et société

# Interdisciplinarité

En quoi  
l'informatique  
est-elle une  
science  
interdisciplinaire ?

## Disciplines

- Mathématiciens-physiciens.
- Chimistes-biologistes.
- Arts-visuels, musiciens, ...

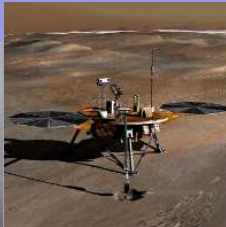
## Nécessité

**Programmation** Pour le plaisir ?

**Base de donnée** Herbiers ou classifications  
d'espèces (biologie) ?

**Graphisme** Traitement de l'image,  
images 3D ?

# Machines



## Informatique- Robotique.

Où s'arrêter ?

En haut à gauche :

Phœnix sur Mars<sup>2</sup>.

En bas :

Aspirateur-récurer iRobot<sup>®</sup>.

# Variétés



Les domaines de l'informatique sont infinis :

- Téléphonie.
- Paiements.
- Gestion des billets.
- Gestion des traffics.
- Analyse de l'information.
- ...



“Rien n’est jamais acquis à l’homme...”.

Science des réseaux, toute informatique se situe dans un *contexte*.

## Exemples

- Sauvegardes chez un particulier ou en entreprise.
- Déploiement d’un ensemble de machines.
- Installation en période de crise économique.
- Pérénnité d’un groupe de programmeurs.

“Rien n’est jamais acquis à l’homme...”.

Science des réseaux, toute informatique se situe dans un *contexte*.

### Exemples

- Sauvegardes chez un particulier ou en entreprise.
- Déploiement d’un ensemble de machines.
- Installation en période de crise économique.
- Pérénnité d’un groupe de programmeurs.

### Difficultés

- Coût du matériel.
- Pérénnité des solutions (obsolescence programmée...).
- Stabilité du système (serveurs web).
- Sécurisation matérielle et logicielle, protection des données.

# Contexte historique

- 1
  - Genève : Logiciels libres.
  - Vaud : MacOS<sup>®</sup>.
  - Neuchatel : Windows<sup>®</sup>.
- 2 Crise économique mondiale et cantonale ⇒  
Crédits zéro en 2010. ⇔
- 3 Compétences présentes en informatiques au lycée :  
LINUX ET MacOS<sup>®</sup>.
- 4 Contexte éducatif et coopératif.

# Contexte historique

- 1
  - Genève : Logiciels libres.
  - Vaud : MacOS<sup>©</sup>.
  - Neuchatel : Windows<sup>©</sup>.
- 2 Crise économique mondiale et cantonale ⇒ Crédits zéro en 2010.
- 3 Compétences présentes en informatiques au lycée : LINUX ET MacOS<sup>©</sup>.
- 4 Contexte éducatif et coopératif.



## Choix : logiciels libres

Système d'exploitation : Linux.

Distribution : Debian ou LinuxMint.

Machines : récupération Dell ou machines montées par l'école d'ingénieurs.

Libre accès : sans risque légaux (piratage), utilisables sur toutes les plates-formes.

# Composants (signification)

Un ordinateur est construit autour des composants suivants :

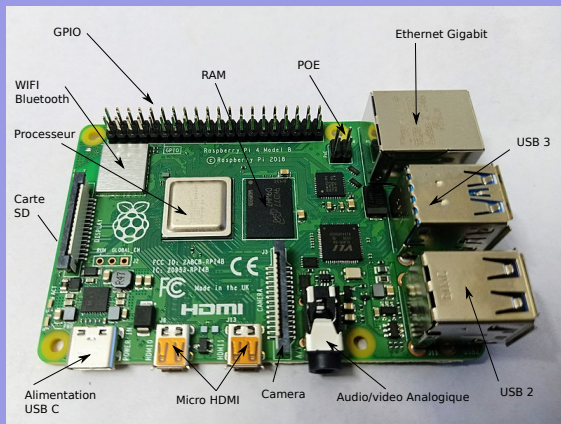
**Processeur** Un circuit électronique qui réalise les calculs,

**Mémoire vive** Une mémoire rapide qui stocke provisoirement les informations et s'efface quand on éteint l'ordinateur,

**Mémoire morte** Une mémoire lente qui stocke les informations et persiste quand l'ordinateur est éteint,

**Interfaces** Des ports permettant de faire entrer et sortir des informations : ports réseau (RJ45, wifi, bluetooth), ports USB, port d'alimentation, entrée/sortie son, connecteurs souris et clavier, connecteurs d'écran (vga, dvi, hdmi).

# Composants (Visualisation)



Référence image<sup>3</sup>

- RAM** Random Access Memory : mémoire vive.
- POE** Power Over Ethernet : alimentation réseau.
- GPIO** General Purpose Input Output : électronique externe.
- SD** Secure Digital : mémoire morte.
- USB** Universal Serial Bus : communications périphériques.
- HDMI** High Definition Multimedia Interface : écrans.
- WIFI** Wireless Fidelity : marque de communication sans fils.
- Bluetooth** Harald à la dent bleue : norme de communication sans fils.
- Ethernet** Net dans l'éther : protocole de communication local.

# Composants (Évaluation)

Les grandeurs caractéristiques des composants sont les suivantes :

**Processeur** caractérisé par sa fréquence de calcul en GHz (Giga Hertz), soit en opérations par seconde :  $3 \text{ GHz} = 3$  milliards d'opération par seconde ; un nombre de coeur (2,4,6,...),

**Mémoire vive** caractérisée par sa capacité en Go (Giga octet) :  $3 \text{ Go} = 3$  milliards d'octets,

**Mémoire morte** caractérisée par sa capacité en Go (Giga octet) :  $500 \text{ Go} = 500$  milliards d'octets,

**Interfaces** caractérisés par leur débit en Go/s : soit en Go par seconde :  $1 \text{ Go/s} =$  un milliard d'octet transféré par seconde.

# Connecteurs

VGA<sup>4</sup>DVI<sup>5</sup>HDMI<sup>6</sup>USB 2<sup>7</sup>USB C<sup>8</sup>RJ45<sup>9</sup>



# Atelier

## Au travail

- 1 Trouvez le nombre de processeur du Raspberrypi4,
- 2 trouvez la fréquence de chacun de ses processeurs,
- 3 trouvez le nombre de millions de calculs qu'il peut réaliser en deux secondes,
- 4 trouvez la mémoire vive dont il dispose,
- 5 trouvez le débit de sa sortie RJ45,
- 6 trouvez le minimum de mémoire morte que son système nécessite et
- 7 À la maison (si vous le voulez et avec précautions) : démontez votre smartphone et faites une reconnaissance de tous ses composants. Est-ce possible dans tous les cas ?

# Atelier

## Au travail

- 1 quadcore, c'est-à-dire **quatre processeurs** ou cœurs,
- 2 trouvez la fréquence de chacun de ses processeurs,
- 3 trouvez le nombre de millions de calculs qu'il peut réaliser en deux secondes,
- 4 trouvez la mémoire vive dont il dispose,
- 5 trouvez le débit de sa sortie RJ45,
- 6 trouvez le minimum de mémoire morte que son système nécessite et
- 7 À la maison (si vous le voulez et avec précautions) : démontez votre smartphone et faites une reconnaissance de tous ses composants. Est-ce possible dans tous les cas ?

# Atelier

## Au travail

- 1 quadcore, c'est-à-dire **quatre processeurs** ou cœurs,
- 2 une fréquence de **1,5 GigaHertz (64 bits)**,
- 3 il peu réaliser **12'000 millions de calculs en deux secondes**,
- 4 trouvez la mémoire vive dont il dispose,
- 5 trouvez le débit de sa sortie RJ45,
- 6 trouvez le minimum de mémoire morte que son système nécessite et
- 7 À la maison (si vous le voulez et avec précautions) : démontez votre smartphone et faites une reconnaissance de tous ses composants. Est-ce possible dans tous les cas ?

# Atelier

## Au travail

- 1 quadcore, c'est-à-dire **quatre processeurs** ou cœurs,
- 2 une fréquence de **1,5 GigaHertz (64 bits)**,
- 3 il peu réaliser **12'000 millions de calculs en deux secondes**,
- 4 sa mémoire vive peut être de **2, 4, 6 ou 8 Go**,
- 5 trouvez le débit de sa sortie RJ45,
- 6 trouvez le minimum de mémoire morte que son système nécessite et
- 7 À la maison (si vous le voulez et avec précautions) : démontez votre smartphone et faites une reconnaissance de tous ses composants. Est-ce possible dans tous les cas ?

# Atelier

## Au travail

- 1 quadcore, c'est-à-dire **quatre processeurs** ou cœurs,
- 2 une fréquence de **1,5 GigaHertz (64 bits)**,
- 3 il peu réaliser **12'000 millions de calculs en deux secondes**,
- 4 sa mémoire vive peut être de **2, 4, 6 ou 8 Go**,
- 5 le **Go/s**,
- 6 trouvez le minimum de mémoire morte que son système nécessite et
- 7 À la maison (si vous le voulez et avec précautions) : démontez votre smartphone et faites une reconnaissance de tous ses composants. Est-ce possible dans tous les cas ?

# Atelier

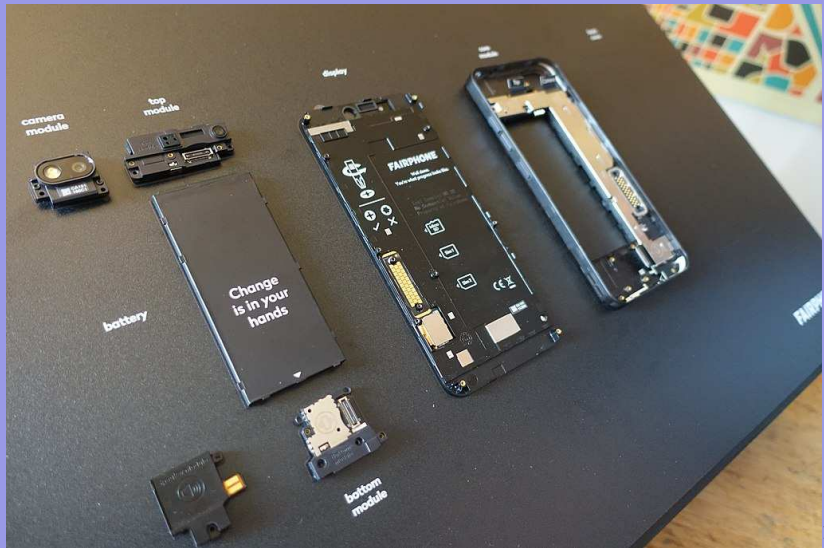
## Au travail

- 1 quadcore, c'est-à-dire **quatre processeurs** ou cœurs,
- 2 une fréquence de **1,5 GigaHertz (64 bits)**,
- 3 il peu réaliser **12'000 millions de calculs en deux secondes**,
- 4 sa mémoire vive peut être de **2, 4, 6 ou 8 Go**,
- 5 le **Go/s**,
- 6 son système d'exploitation nécessite **dix Go**.
- 7 À la maison (si vous le voulez et avec précautions) : démontez votre smartphone et faites une reconnaissance de tous ses composants. Est-ce possible dans tous les cas ?

# Atelier

## Au travail

- ① quadcore, c'est-à-dire **quatre processeurs** ou cœurs,
- ② une fréquence de **1,5 GigaHertz (64 bits)**,
- ③ il peu réaliser **12'000 millions de calculs en deux secondes**,
- ④ sa mémoire vive peut être de **2, 4, 6 ou 8 Go**,
- ⑤ le **Go/s**,
- ⑥ son système d'exploitation nécessite **dix Go**.
- ⑦ Certains smartphones ne permettent pas de retirer la batterie pour voir les composants, par exemple. Voyez le cas du Fairphone qui est entièrement démontable.





# Systèmes d'exploitations

## OS : operating system

Ce qui fait fonctionner l'ordinateur ; interface entre le matériel et l'utilisateur.

## Principaux OS

### Windows

Installé par défaut  
ou acheté en pack  
Payant

### MacOS®

Installé par défaut  
sur machine Apple  
Payant

### Linux®

De type Unix  
installable "partout"  
Gratuit

# Systèmes d'exploitation propriétaires

## Mac OS<sup>©</sup>

Originalité : Concepteur d'ordinateur (hardware) et de système d'exploitation (software).

Depuis Mac OS X, système de type UNIX très proche de Linux. Seule l'interface graphique est propre à Macintosh. Beaucoup des logiciels intégrés sont des logiciels libres : apache, samba, libreoffice, gimp, ...

## Windows<sup>©</sup>

Windows<sup>©</sup> préinstallé de force beaucoup d'ordinateurs. Soumis à licence payante, il existe en version serveur, mais est assez instable pour que Microsoft utilise linux pour son cloud Azure.

# Binaire - Source

## Binaire

Un ordinateur ne comprend que les zéros et uns dont est composé un fichier binaire.

```
0100010 1001010 01010001  
10010001010000101
```

## Source

Un programme s'écrit dans un langage compréhensible par l'homme et est traduit par un compilateur en binaire.

```
if (argent = bonheur)  
  then (1/3 population = triste)
```

Exemple de l'utilité de l'ouverture du code source : [Projet télescopique](#)

# Licences

## Licences propriétaires

Les logiciels peuvent uniquement être utilisés.

Ils ne peuvent pas être étudiés.

Ils restent la propriété de leur concepteurs.

## Licences libres

Le code des logiciels peut être étudié.

Il peut être modifié, amélioré et réutilisé.

Le code appartient à la communauté.

Propriétaires ou libres, il existe beaucoup de type de licences.

Dans le domaine du libre, les plus connues sont la GPL (licence publique générale) et les Creative Commons. Relevons aussi que Windows<sup>®</sup> et MacOS<sup>®</sup> sont sous licence propriétaire, malgré l'utilisation par Mac d'un micronoyau Mach enrichi par un noyau BSD (licence BSD).

# Free Software Foundation (FSF-GNU), licences libres

## Quatre libertés

- Liberté 0** La liberté d'exécuter le programme — pour tous les usages ;
- Liberté 1** La liberté d'étudier le fonctionnement du programme — ce qui suppose l'accès au code source ;
- Liberté 2** La liberté de redistribuer des copies — ce qui comprend la liberté de vendre des copies ;
- Liberté 3** La liberté d'améliorer le programme et de publier ses améliorations — ce qui suppose, là encore, l'accès au code source.

# Libertés

- La liberté 3 encourage la création d'une communauté de développeurs améliorant le logiciel et permet le fork.
- « Libre » ne doit pas être compris comme « gratuit ». Chacun a le droit de redistribuer gratuitement ou non un logiciel libre.
- Ces libertés doivent être irrévocables.
  - possibilité d'en jouir sans devoir prévenir un tiers ;
  - possibilité de redistribuer le programme sous toute forme, notamment compilée ;
  - le code source doit être accessible pour jouir des libertés d'étude et d'amélioration ;
  - possibilité de fusionner des logiciels libres dont on n'est pas soi-même l'auteur<sup>11</sup>.

# GNU-Linux

**GNU** Gnu is Not Unix. Richard Stallman crée la première licence libre. Il n'admettait pas de disposer d'une imprimante dont le logiciel ne fonctionnait pas, mais qu'il lui était interdit de corriger pour des raisons de licence.

**Linux** Linus Torvalds crée le noyau de ce qu'il nomme Lin-u-x. Il s'agit d'un noyau créé de tout pièces sur le modèle des systèmes d'exploitation UNIX de l'époque.

**GNU-Linux** est constitué d'un noyau monolithique et d'un ensemble de logiciels libres.

# Distributions Linux<sup>©</sup>

## Base .deb

**Debian** Système totalement libre : refus d'intégrer du code source sous licence non libre.

**Ubuntu** Système dérivé de Debian, mais se permettant si nécessaire d'intégrer du code non libre.

**LinuxMint** Système dérivé de Ubuntu.

## Base .rpm

**Redhat** Système orienté entreprises, payant.

**Mandrake** Système d'origine française.

**OpenSuSe** Système d'origine allemande.



# Logiciels

*Quelques exemples multi-plateforme et libres*

**Texte** OpenOffice  
(fork  
libreoffice),  
Scribus,  
Latex, ...

**Image** Gimp,  
Digikam,  
ImageMagik,  
...

**Code** Perl, Python,  
C, ...

**Web** Firefox,  
Apache,  
Wordpress,  
...

**Bases** MariaDB, ...

**Astro** Celestia,  
Kstar,  
Stellarium,  
...

**Mail** Thunderbird,  
Kmail,  
Evolution,  
...

# Services

## Serveur

Une machine allumée en permanence.

## Services

Des logiciels toujours accessibles comme :

Serveur Web Internet

Serveur DNS Nom et adresse

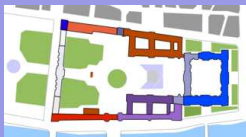
Serveur de temps Date et heure

Serveur mail Mails

Serveur de fichiers Fichiers

Serveur de streaming Films

# Réseau



12

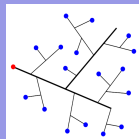
## Réseau d'habitations

**Transport** Route, air, eau,  
carrefour

**Moyen** Voiture, vélo, pieds

**Lieu** Adresse, numéro

**Ouvertures** Porte, fenêtre, boîte  
aux lettres



13

## Réseau de machines

**Transport** Ethernet, wifi,  
routeurs

**Protocole (Moyen)** Http, ftp,  
ssh, smb, ...

**Lieu** Nom de domaine  
ou (DNS) adresse  
IPV4 : 192.168.0.1

**Ouvertures** Ports :80, 22, ...

# Atelier

## Au travail

- Adresse IP** Déterminez l'adresse IP de votre machine,
- Réseau** comparez votre adresse IP à celle de vos camarades,
- Transport** tracez la route de vos données vers Raspinoir,
- Protocole** connectez-vous en http à l'adresse de Raspinoir ;  
qu'obtenez-vous ?  
Connectez-vous en ssh à cette même adresse ;  
qu'obtenez-vous ?  
Connectez-vous sur le port FTP à cette adresse ;  
qu'obtenez-vous ?  
Connectez-vous au serveur de vidéo de la physique à  
l'aide de VLC et commencez à visionner un film.  
Quel est le protocole utilisé ?

# Atelier

## Au travail

**Adresse IP** On utilise la commande **ip addr**

**Réseau** comparez votre adresse IP à celle de vos camarades,

**Transport** tracez la route de vos données vers Raspinoir,

**Protocole** connectez-vous en http à l'adresse de Raspinoir ;  
qu'obtenez-vous ?

Connectez-vous en ssh à cette même adresse ;  
qu'obtenez-vous ?

Connectez-vous sur le port FTP à cette adresse ;  
qu'obtenez-vous ?

Connectez-vous au serveur de vidéo de la physique à  
l'aide de VLC et commencez à visionner un film.

Quel est le protocole utilisé ?

# Atelier

## Au travail

**Adresse IP** On utilise la commande `ip addr`

**Réseau** les trois **premier nombres sont identiques** ; ils définissent le réseau du lbc,

**Transport** tracez la route de vos données vers Raspinoir,

**Protocole** connectez-vous en http à l'adresse de Raspinoir ; qu'obtenez-vous ?

Connectez-vous en ssh à cette même adresse ; qu'obtenez-vous ?

Connectez-vous sur le port FTP à cette adresse ; qu'obtenez-vous ?

Connectez-vous au serveur de vidéo de la physique à l'aide de VLC et commencez à visionner un film.

Quel est le protocole utilisé ?

# Atelier

## Au travail

**Adresse IP** On utilise la commande **ip addr**

**Réseau** les trois **premier nombres sont identiques** ; ils définissent le réseau du lbc,

**Transport** On utilise la commande **traceroute IP ou nom de domaine**

**Protocole** connectez-vous en http à l'adresse de Raspinoir ; qu'obtenez-vous ?  
Connectez-vous en ssh à cette même adresse ; qu'obtenez-vous ?  
Connectez-vous sur le port FTP à cette adresse ; qu'obtenez-vous ?  
Connectez-vous au serveur de vidéo de la physique à l'aide de VLC et commencez à visionner un film.  
Quel est le protocole utilisé ?

# Atelier

## Au travail

**Adresse IP** On utilise la commande **ip addr**

**Réseau** les trois **premier nombres sont identiques**; ils définissent le réseau du lbc,

**Transport** On utilise la commande **traceroute IP ou nom de domaine**

**Protocole** Le **site internet** principal de Raspinoir.  
Une **interface de connexion** sur Rapinoir.  
**Une erreur**, car aucun serveur FTP n'est présent.  
**Une erreur, car il est éteint** mais le protocole est UPNP/DLNA



# Notions générales de sécurité

**Machines** Fermeture des ports inutilisés (fermeture des portes)

**Logiciels** Mises-à-jour régulières et utilisation des dépôts officiels

**Piratage** Hameçonnage : demande par mail de modif. de coordonnées bancaires, par exemple, manipulation d'URL :

`https://www.lbc.ch/?1e1bc.ch,`

virus, pièces jointes, macros, javascript, html

mots de passe (principal navigateur), force brute, dictionnaires, ...

## Chiffrement

Chiffrement - déchiffrement, cryptage, authentification : principes<sup>14</sup>

## Chiffrement asymétrique

BOB

*Je t'aime*

Condensat

```
MuclffyyE+rP6WUDhvL
ZXviLePue+FLgU83Wfy
EoQJBA0/c8xMdbJgmjC
yxw1Zw0nyz05Vt/0C0Q
KGQpR086WPI9ZvIziXQ
H06qw11YTD1rSCcUG1i
```

Chiffrement  
Clef privée  
Bob

```
md+HTRdJ5Q7tbIGoERSG
VTQpFuTsJPTWuQsTsAW
zaFX06y0B12bAgMBAEC
dTUCvtkCAp+HYTT9yHD
9I9zcbAHcwS8K0WabIKc
DzJncUu2tfGVcPDxD+ZE
```

Chiffrement  
Clef publique  
Alice

```
v80Rcblc3W/AL3a0BntE
xrLnP5vmdYg51BjalvJy
ISCENAhDzDkKW43AefIK
Ewx6HgVlBgPpXIazcdIH
WaE7CIPuk9pxar0DzADS
xzh2beww/1GGY+CyFu2N
```

Envoi

ALICE

*Je t'aime*

Décondensat

```
MuclffyyE+rP6WUDhvL
ZXviLePue+FLgU83Wfy
EoQJBA0/c8xMdbJgmjC
yxw1Zw0nyz05Vt/0C0Q
KGQpR086WPI9ZvIziXQ
H06qw11YTD1rSCcUG1i
```

Authentification  
Clef publique  
Bob

```
md+HTRdJ5Q7tbIGoERSG
VTQpFuTsJPTWuQsTsAW
zaFX06y0B12bAgMBAEC
dTUCvtkCAp+HYTT9yHD
9I9zcbAHcwS8K0WabIKc
DzJncUu2tfGVcPDxD+ZE
```

Déchiffrement  
Clef privée  
Alice

```
v80Rcblc3W/AL3a0BntE
xrLnP5vmdYg51BjalvJy
ISCENAhDzDkKW43AefIK
Ewx6HgVlBgPpXIazcdIH
WaE7CIPuk9pxar0DzADS
xzh2beww/1GGY+CyFu2N
```

# Chiffrage

## Chiffrage - déchiffrage : PGP

### AUTODÉFENSE COURRIEL

La surveillance de masse viole nos droits fondamentaux et fait planer un risque sur notre liberté de parole.

**Mais nous sommes loin d'être désarmés face à cette situation.**



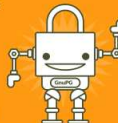
Le mot de passe protégeant votre compte de courriel n'est qu'une fine couche de sécurité qui ne peut vous protéger contre les attaques de systèmes de surveillance sophistiqués.

En route vers sa destination, chaque courriel transite par plusieurs systèmes informatiques. Les agences de surveillance en profitent pour lire des millions et des millions de courriels.

Même si vous n'avez rien à cacher, rappelez-vous qu'en envoyant un courriel normal, vous exposez aussi vos correspondants.

### Récupérez votre vie privée avec GnuPG !

Tout ce dont vous avez besoin est un simple programme appelé **GnuPG**. Il chiffre votre courriel en un code que seul le destinataire peut lire.



GnuPG fonctionne sur presque tous les ordinateurs et smartphones. Il est sous licence libre et ne coûte rien. Chaque utilisateur possède une paire unique **clef publique - clef privée**. Ces clefs sont des suites de nombres aléatoires.



#### VOTRE CLEF PUBLIQUE

Votre clef publique n'est pas comme une clef physique, parce que vous la partagez. Elle est stockée sur un serveur en ligne, où chacun peut la rechercher et la télécharger.

Cette clef est utilisée, avec GnuPG, pour chiffrer les courriels qui vous seront envoyés.



#### VOTRE CLEF PRIVÉE

Votre clef privée ressemble plus à une clef physique, car vous la conservez avec vous (sur votre ordinateur).

À l'aide de GnuPG et de votre clef privée, vous pouvez décoder les courriels chiffrés qui vous sont envoyés.



# Chiffrage

## Chiffrage - déchiffrage : PGP



GnuPG est un **logiciel sous licence libre** ; il est totalement transparent et tout le monde peut le copier ou faire sa propre version. Cela le rend plus sûr vis-à-vis de la surveillance qu'un logiciel propriétaire (comme Windows ou Word).  
Plus d'infos sur [FSF.org](http://FSF.org)

Pour nous protéger de la surveillance, nous devons apprendre à utiliser GnuPG et **échanger nos clés publiques** en même temps que nos adresses de courriel.

Des milliers de personnes utilisent déjà GnuPG, des activistes, des journalistes, des lanceurs d'alerte ou des gens ordinaires. Chaque personne l'utilisant renforce notre communauté et montre aux agences de surveillance que nous sommes prêts à la riposte.

Apprenez l'autodéfense courriel. Abordez GnuPG en 30 minutes sur [EmailSelfDefense.FSF.org/fr](http://EmailSelfDefense.FSF.org/fr)

Copyright 2014 Free Software Foundation.  
**Remix encouragé ! Prenez la source à l'URL ci-dessus.**

Conception de l'infographie et du guide par **Journalism++** - Traduction par **Framasoft**

Si un courriel chiffré avec GnuPG tombe entre de mauvaises mains, il semblera incompréhensible. Sans la clé privée du vrai destinataire, il est presque impossible de le lire.

Pour le vrai destinataire, il s'ouvre comme un courriel normal. Facile !

L'expéditeur et le destinataire sont maintenant en sécurité. Même si ce courriel ne contient rien de privé, le chiffrement entrave les systèmes de surveillance de masse.  
**Va te faire voir, surveillance !**

Voir l'excellent mode d'emploi d'« Autodéfense courriel »<sup>15</sup>.

# Chiffrage

## Application : communications client-serveur

web : https

terminal : ssh

mails : pops, imaps : ssl/tls ; gpg

fichiers : scp, sftp

## Application : navigateur sécurisé-anonyme

Navigateur TOR : recommandé par Edward Snowden.

Principe des poupées russes. Un excellent article pour comprendre avec des mots http, https et Tor : <https://docs.framasoft.org/fr/comprendre/https-vpn.html>

# Table des matières

## 2 DF Informations et données

- Stockage : binaire
- Texte, codage, encodage
- Formats de fichier
- Bases de données

## 1 DF Systèmes numériques

## 2 DF Informations et données

## 3 DF Algo. et programmation

## 4 DF Informatique et société

# Représentation d'un nombre

Comment un ordinateur comprend-il les nombres ?

# Représentation d'un nombre

Comment un ordinateur comprend-il les nombres ?

Représentation décimale :

127 =



# Représentation d'un nombre

Comment un ordinateur comprend-il les nombres ?

Représentation décimale :

$$127 = 100 + 20 + 7 =$$

# Représentation d'un nombre

Comment un ordinateur comprend-il les nombres ?

Représentation décimale :

$$127 = 100 + 20 + 7 = 1 \cdot 100 + 2 \cdot 10 + 7 \cdot 1 =$$

# Représentation d'un nombre

Comment un ordinateur comprend-il les nombres ?

Représentation décimale :

$$127 = 100 + 20 + 7 = 1 \cdot 100 + 2 \cdot 10 + 7 \cdot 1 = 1 \cdot 10^2 + 2 \cdot 10^1 + 7 \cdot 10^0$$

# Représentation d'un nombre

Comment un ordinateur comprend-il les nombres ?

Représentation décimale :

$$127 = 100 + 20 + 7 = 1 \cdot 100 + 2 \cdot 10 + 7 \cdot 1 = 1 \cdot 10^2 + 2 \cdot 10^1 + 7 \cdot 10^0$$

Représentation binaire :

# Représentation d'un nombre

Comment un ordinateur comprend-il les nombres ?

Représentation décimale :

$$127 = 100 + 20 + 7 = 1 \cdot 100 + 2 \cdot 10 + 7 \cdot 1 = 1 \cdot 10^2 + 2 \cdot 10^1 + 7 \cdot 10^0$$

Représentation binaire :

$$1101 =$$

# Représentation d'un nombre

Comment un ordinateur comprend-il les nombres ?

Représentation décimale :

$$127 = 100 + 20 + 7 = 1 \cdot 100 + 2 \cdot 10 + 7 \cdot 1 = 1 \cdot 10^2 + 2 \cdot 10^1 + 7 \cdot 10^0$$

Représentation binaire :

$$1101 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 =$$

# Représentation d'un nombre

Comment un ordinateur comprend-il les nombres ?

Représentation décimale :

$$127 = 100 + 20 + 7 = 1 \cdot 100 + 2 \cdot 10 + 7 \cdot 1 = 1 \cdot 10^2 + 2 \cdot 10^1 + 7 \cdot 10^0$$

Représentation binaire :

$$1101 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 8 + 4 + 0 + 1 = 13$$

# A l'envers ? Première méthode.

Il s'agit de passer du nombre décimal 13 à son correspondant binaire 1101. Deux méthodes sont possibles.

La première est intuitive :

- 1 On trouve la plus grande puissance de 2 inférieure ou égale au nombre décimal, ici  $2^3$  car  $2^3 = 8 < 13$ ,
- 2 On calcule le reste, ici  $13 - 8 = 5$  et on reprend le premier point pour celui-ci et les suivants.

On obtient alors  $2^2 = 4$  et il reste  $1 = 2^0$ . Le nombre binaire cherché se factorise donc ainsi :

$$13 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$$

Le résultat est donc donné par les facteurs de chaque puissance de deux : 1101.



# A l'envers ? Seconde méthode.

Il s'agit de passer du nombre décimal 13 à son correspondant binaire 1101.

La seconde méthode est plus "mécanique" :

elle consiste à diviser le nombre par deux successivement.

Les restes constituent le nombre binaire à l'envers.

Par exemple,  $13/2=6$  reste 1. Puis,  $6/2=3$  reste 0. Puis,  $3/2=1$  reste 1 et enfin,  $1/2=0$  reste 1.

Les restes sont successivement 1011. En inversant, on obtient le nombre binaire 1101 qui correspond au nombre décimal 13.

# Exercices

Convertissez les nombres suivants :

- 8 en binaire
- 21 en binaire
  
- 1001101 en décimal
  
- 1101,101 en décimal
  
- 14,25 en binaire

# Exercices

Convertissez les nombres suivants :

- 8 en binaire = 1000 car  $1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 8$
- 21 en binaire
- 1001101 en décimal
- 1101,101 en décimal
- 14,25 en binaire

# Exercices

Convertissez les nombres suivants :

- 8 en binaire = 1000 car  $1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 8$
- 21 en binaire = 10101 car  
 $1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 21$
- 1001101 en décimal
- 1101,101 en décimal
- 14,25 en binaire

# Exercices

Convertissez les nombres suivants :

- 8 en binaire = 1000 car  $1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 8$
- 21 en binaire = 10101 car  
 $1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 21$
- 1001101 en décimal = 77 car  
 $1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 77$
- 1101,101 en décimal
  
- 14,25 en binaire

# Exercices

Convertissez les nombres suivants :

- 8 en binaire = 1000 car  $1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 8$
- 21 en binaire = 10101 car  
 $1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 21$
- 1001101 en décimal = 77 car  
 $1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 77$
- 1101,101 en décimal = 13,625 car  
 $1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} = 13,625$
- 14,25 en binaire

# Exercices

Convertissez les nombres suivants :

- 8 en binaire = 1000 car  $1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 8$
- 21 en binaire = 10101 car  
 $1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 21$
- 1001101 en décimal = 77 car  
 $1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 77$
- 1101,101 en décimal = 13,625 car  
 $1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} = 13,625$
- 14,25 en binaire = 1110,01 car  
 $1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} = 14,25$

# Byte, bit et octet

## Décimal

0, 1, 2, ..., 9 : chiffres.

23, 45, 234, ... : nombres.



# Byte, bit et octet

## Décimal

0, 1, 2, ..., 9 : chiffres.

23, 45, 234, ... : nombres.

## Binaire

0 et 1 : **bits**.

101, 11, 10, ... : nombres.

# Byte, bit et octet

## Décimal

0, 1, 2, ..., 9 : chiffres.

23, 45, 234, ... : nombres.

## Binaire

0 et 1 : **bits**.

101, 11, 10, ... : nombres.

(Attention, 101 binaire ne se lit pas "cent un" et ne correspond pas à la valeur décimale 101. En réalité, on a :

$0_b = 0_d$  ;  $1_b = 1_d$  ;  $10_b = 2_d$  ;  $11_b = 3_d$  ;  $100_b = 4_d$  ;  $101_b = 5_d$  ...)

# Byte, bit et octet

- Avec un *bit*, on a les nombres 0 et 1, soit deux nombres.

# Byte, bit et octet

- Avec un *bit*, on a les nombres 0 et 1, soit deux nombres.
- Avec deux *bits*, on a les nombres 00, 01, 10 et 11, soit quatre nombres.

# Byte, bit et octet

- Avec un *bit*, on a les nombres 0 et 1, soit deux nombres.
- Avec deux *bits*, on a les nombres 00, 01, 10 et 11, soit quatre nombres.
- Avec trois *bits*, on a les nombres 000, 001, 010, 011, 100, 101, 110 et 111, soit huit nombres.

# Byte, bit et octet

- Avec un *bit*, on a les nombres 0 et 1, soit deux nombres.
- Avec deux *bits*, on a les nombres 00, 01, 10 et 11, soit quatre nombres.
- Avec trois *bits*, on a les nombres 000, 001, 010, 011, 100, 101, 110 et 111, soit huit nombres.
- Avec quatre *bits*, on a ... ?

# Byte, bit et octet

- Avec un *bit*, on a les nombres 0 et 1, soit deux nombres.
- Avec deux *bits*, on a les nombres 00, 01, 10 et 11, soit quatre nombres.
- Avec trois *bits*, on a les nombres 000, 001, 010, 011, 100, 101, 110 et 111, soit huit nombres.
- Avec quatre *bits*, on a ... ?
- Avec  $n$  *bits*, on a un **arrangement avec répétition de  $k$  éléments choisis parmi deux** (0 ou 1). Les mathématiques donnent alors le nombre de ces arrangements :

$$\bar{A}_k^2 = 2^k$$

# Byte, bit et octet

Avec huit bits, on a donc :

$$\bar{A}_8^2 = 2^8 = 256 \text{ possibilités}$$

On peut donc représenter 256 nombres.

Un **octet** est un nombre de huit bits.  
Un *octet* est aussi nommé **byte**.



# kilobyte, Megabyte, Gigabyte, ...

## Décimal

kilo  $\equiv$  nombre de possibilités de réaliser un nombre à l'aide de 3 chiffres :

$$\bar{A}_3^{10} = 10^3 = 1000 \text{ possibilités}$$

# kilobyte, Megabyte, Gigabyte, ...

## Décimal

kilo  $\equiv$  nombre de possibilités de réaliser un nombre à l'aide de 3 chiffres :

$$\bar{A}_3^{10} = 10^3 = 1000 \text{ possibilités}$$

## Binaire

kilo  $\equiv$  nombre de possibilités de réaliser un nombre à l'aide de 10 bits :

$$\bar{A}_{10}^2 = 2^{10} = 1024 \text{ possibilités}$$

# kilobyte, Megabyte, Gigabyte, ...

- 1 Un kilobyte ou kilooctet, noté ko, correspond à 1024 octets.
- 2 Un Megabyte ou Megaoctet, noté Mo, correspond à 1024 kilooctet, soit 1'048'576 bytes ou octets.
- 3 Un Gigabyte ou Gigaoctet, noté Go, correspond à 1024 Megaoctet, soit 1'048'576 kilobyte ou 1'073'741'824 bytes.

L'ordre de grandeur correspond à la notation décimale :

- 1 kilo pour milliers.
- 2 Mega pour millions.
- 3 Giga pour milliards.

Un processeur peut être comparé à une machine à calculer dont l'affichage comporte un nombre limité de chiffres.

## En décimal

Pour prendre un exemple simple, supposons que notre calculatrice ne peut calculer qu'avec trois chiffres.

Le nombre de nombres entiers positifs différents qu'elle peut afficher est de  $10^3$ . En effet, on comprend bien que ceux-ci varient de 000 à 999. Ils sont donc au nombre de 1000. Le plus petit est 000 et le plus grand 999.

Si une place est réservée au signe, alors le nombre de nombres possible est de  $10^2$  nombres positifs (de +00 à +99) et de  $10^2-1$  nombres négatifs (de -01 à -99), soit au total  $100 + 99 = 199$ .

Un processeur peut être comparé à une machine à calculer dont l'affichage comporte un nombre limité de chiffres.

## En binaire

Entiers positifs sur trois bits : \_ \_ \_

000 → 0

001 → 1

010 → 2

011 → 3

100 → 4

101 → 5

110 → 6

111 → 7

Au total  $2^3=8$  chiffres

Minimum : 000 -> 0

Maximum : 111 ->

$$1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 4 + 2 + 1 = 7$$

Un processeur peut être comparé à une machine à calculer dont l'affichage comporte un nombre limité de chiffres.

## En binaire

Entiers signé sur trois bits : \_ \_ \_

000 → 0

001 → 1

010 → 2

011 → 3

100 → -0

101 → -1

110 → -2

111 → -3

Premier bit : signe

$(-1)^0 = +1 = +$  ;  $(-1)^1 = -1 = -$

Au total  $2^3=8$  chiffres

4 chiffres positifs, 3 chiffres négatifs

Minimum : 111 -> -3

Maximum : 011 -> 3

# Exercices

Sur un processeur 16 bits, combien de nombres entiers non signés peut on représenter ?

(un processeur 16 bits peut manipuler des “mots” de 16 bits, exemple 1110110000110110)

# Exercices

Sur un processeur 16 bits, combien de nombres entiers non signés peut on représenter ?

(un processeur 16 bits peut manipuler des “mots” de 16 bits, exemple 1110110000110110)

On a  $2^{16} = 65'536$  possibilités, soit autant de nombres.



# Exercices

Sur le même processeur, quels sont les plus petits et plus grands nombres entiers signés qui puissent être représentés ?  
(c'est à dire avec la possibilité de faire des nombres positifs et négatifs)

# Exercices

Sur le même processeur, quels sont les plus petits et plus grands nombres entiers signés qui puissent être représentés ?

(c'est à dire avec la possibilité de faire des nombres positifs et négatifs)

On a un bit de signe (0 ou 1) et 15 bits pour le nombre, soit  $2^{15} = 32'768$  possibilités. On peut donc représenter des nombres de  $-32'768$  à  $32'767$  (à cause du zéro).

En effet, le nombre le plus élevé est  $0111111111111111$ , soit  $2^{15} + 2^{14} + \dots + 2^1 + 2^0 = 32'767$  et le plus faible  $1111111111111111$ , soit  $-32'767$ .

Mais le nombre  $1000000000000000$  pourrait être utilisé pour  $-32'768$  à la place de  $-0$ .

# Exercices

Sur un processeur 32 bits, si on met dans un fichier une fois tous les nombres entiers non signés, quel sera en Go la taille du fichier ?

# Exercices

Sur un processeur 32 bits, si on met dans un fichier une fois tous les nombres entiers non signés, quel sera en Go la taille du fichier ?

Avec un tel processeur on a  $2^{32} = 4'294'967'296$  nombres possibles (avec le zéro). Chaque nombre est codé sur 32 bits, soit au total  $4'294'967'296 \cdot 32 = 1,37 \cdot 10^{11}$  bits.

Cela représente  $1,37 \cdot 10^{11} / 8 = 1,71 \cdot 10^{10}$  octets ou  $1,71 \cdot 10^{11} / 1024 = 16'777'216$  ko, ou encore  $16'777'216 / 1024 = 16'384$  Mo, c'est-à-dire 16 Go.

# Supports de données



16



17



18



19



20

**Bande magnétique** plastique et oxyde magnétique (oxyde ferrique)  
(haute capacité, accès très lent)

**Disque dur** disque magnétique (haute capacité, accès lent)

**Disque dur ssd** mémoire EEPROM (moyenne capacité, accès rapide)

**Clé USB** mémoire EEPROM (moyenne capacité, accès moyen)

**Carte SD** mémoire EEPROM (faible capacité, accès varié)

# Texte et codage

## Caractères $\leftrightarrow$ table des caractères

### Norme ISO (International Organization for Standardization)

Dérivée des 128 caractères codés de 0 à 127, soit en binaire de 0000000 à 1111111 du codage ASCII (merican Standard Code for Information Interchange).

Huit bit, soit un octet ; correspond à l'ISO-8859-1 ou latin 1. Codage jusqu'à 256 caractères.

Le latin-0 et le latin-1, ISO 8859-1 et ISO 8859-15, décrivent les caractères de l'Europe de l'ouest, avec ou sans le caractère de l'euro.

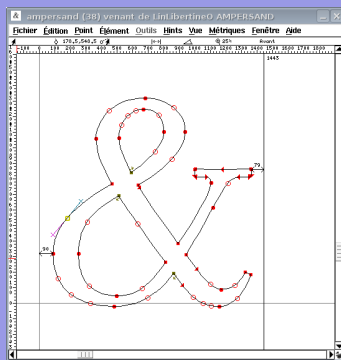
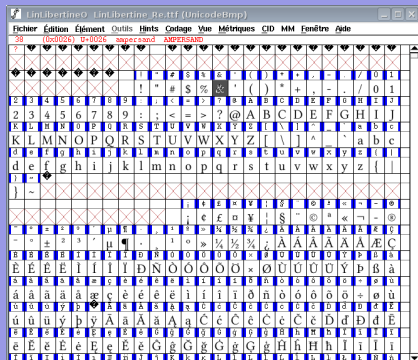
### Norme UTF (Universal coded character set Transformation Format)

#### Table de codage unique : "Unicode".

Elle sépare la représentation du caractère du codage lui-même par l'intermédiaire d'un index numérique nommé *point de code*. Celui-ci s'écrit `U+nombre_hexadécimal`. Les caractères correspondant au latin de base se trouvent entre les points de code 0000 et 007F, ce qui correspond en décimal à un index de 0 à 127, c'est-à-dire aux caractères ASCII.

# Codage et glyphes

A un caractère correspond un seul point de code, mais peut correspondre plusieurs glyphes.



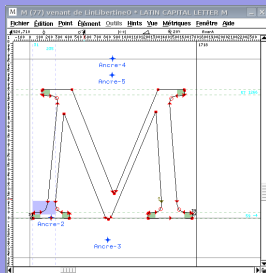
# Fontes – bézier – bitmap

Fontes vectorielles – Fontes bitmap

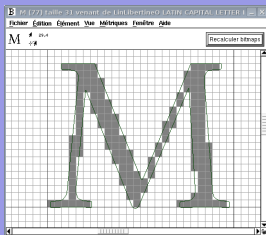
Postscript, truetype – bdf

Un fichier, plusieurs fontes anti-aliasées – Un fichier, une seule fonte non anti-aliasée

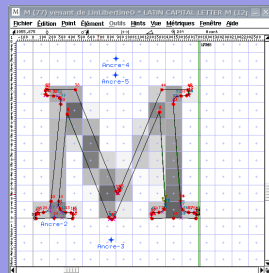
Courbe de bézier



Représentation  
bitmap



Anti-aliasing et  
hinting



Autres problèmes rencontrés : gestion des créneaux (ex. : Tc).



# Moteur de rendu de fontes

Le rendu des fontes se fait automatiquement au moment de la frappe de chaque caractère !!!

Reste à créer vos propre glyphs :  
avec ou sans empattement – avec ou sans serif – à taille fixe ou pas

AaBbCcFcTeFi AaBbCcFcTeFi

AaBbCcFcTeFi

# Structure d'un fichier LibreOffice : odt

## Au travail

- 1 Faites une maquette de documentation de projet avec LibreOffice : titre, entête, pied de page, pagination, table des matières, sections, sous-sections, images ...
- 2 Enregistrez celle-ci dans un répertoire nommé *Maquette* avec le nom de fichier *maMaquette.odt*.
- 3 Faites une copie de ce fichier.
- 4 Renommez ce fichier (pas la copie) *maMaquette.zip*.
- 5 Par un clic droit sur ce dernier, sélectionnez *Extraire ici*.
- 6 Rendez-vous dans le répertoire ou dossier créé.
- 7 Que contient-il ? En déduire ce qu'est un fichier zip.

# Structure d'un fichier LibreOffice : odt

LibreOffice avec styles

## Au travail

- 1 Que contient le répertoire *images*
- 2 Trouvez le fichier *content.xml*. Ouvrez-le avec un éditeur de texte simple (pas LibreOffice). Que contient-il ?
- 3 Trouvez-y une phrase que vous avez écrite et changez là, sans rien toucher d'autre. Enregistrez.
- 4 Sélectionnez tous les fichiers du répertoire, à l'exception du fichier *maMaquette.zip* et cliquez-droit sur ceux-ci.
- 5 Sélectionnez *Compresser* et dans la boîte de dialogue spécifiez *zip* comme format de fichier et enregistrez ce fichier sur le bureau, sous le nom *maSecondeMaquette.zip*.
- 6 Changez l'extension de ce dernier fichier en *odt* et ouvrez-le avec LibreOffice. Que constatez-vous ?

# types de fichiers

Il existe trois types de fichiers :

**les fichiers binaires** Il faut un utilitaire spécial pour voir le contenu binaire. Il s'agit de 0 ou 1 ou éventuellement de caractères hexadécimaux.

**les fichiers types** Plusieurs applications peuvent les ouvrir et les lire. Par exemple, jpg pour les images, mp3 pour les sons, ogg pour les vidéos.

**les fichiers d'application** Seul l'application ayant créé le fichier peut l'ouvrir. Par exemple, xcf pour Gimp ou psd pour Photoshop : dans les deux cas c'est une image, mais les modifications faites dans ces logiciels sont stockées de manière particulière à ceux-ci.

# Tableur LibreOffice : classification d'espèces d'arbres

## Au travail

Avec le tableur de LibreOffice (Calc), réalisez un système qui permet de dénombrer le nombre d'individus (arbres) de plusieurs espèces (Érable, Hêtre, Chêne, sapin, épicéa, ...) sur une région donnée à une date donnée, pour deux observateurs référencés, chacun ayant son propre tableau de remplissage sur la même feuille.

Dans un autre tableau doit se trouver la présentation des espèces et le nombre d'individus total recensés, le nombre total d'individus et le pourcentage d'individus par espèce. Enfin, un tableau supplémentaire doit permettre de voir le nombre d'arbres comptabilisé par chaque observateur.

# Résultat

RecensementArbres.ods - LibreOffice Calc

Fichier Édition Affichage Insertion Format Styles Feuille Données Outils Fenêtre Aide

Liberation Sans 10 % 00

H21

	A	B	C	D	E	F	G	H	I	J
1	Observateur	Vincent				Observateur	Céline			
2	Espèces	Nombre	Région	Date		Espèces	Nombre	Région	Date	
3										
4	Érable	3	Parc Gallet	22.02.00		Érable	4	EPO	08.04.18	
5	Hêtre	5	Parc Crétets	15.04.05		Hêtre	3	EPO	06.09.21	
6	Chêne	3	Téleski	31.05.12		Chêne	6	Gentiannes	14.12.16	
7	Sapin	2	Piscine	41.14.30		Sapin	7	Cernil-Antoine	23.07.20	
8	Épicéa	6	Aéroport	31.02.19		Épicéa	9	Parc Musée	05.05.05	
9										
10	Espèces	Nombre	Pourcentage			Observateur	Nombre			
11										
12	Érable	7	14.6%			Céline	29			
13	Hêtre	8	16.7%			Vincent	19			
14	Chêne	9	18.8%							
15	Sapin	9	18.8%							
16	Épicéa	15	31.3%							
17										
18	Total	48								
19										

## Tableur -> base de donnée

Différents problèmes se présentent :

- Ajout d'espèces à plusieurs places, ajout d'observateurs, ajout de régions ?
- Redondance des deux tableaux des observateurs.
- Présentation complexe.
- Accès à distance avec présentation interface d'acquisition propre à chaque utilisateur, centralisation des données et visualisation de certaines d'entre elles.

## Bases de donnée

Une base de donnée est constituée de deux éléments :

- 1 un ensemble d'informations et
- 2 un structure permettant leur accès.

## Exemples

**Fichier texte CSV** Fichier très simple, structuré en colonnes délimitées par des séparateurs de différents types : tabulations, espaces, virgules, ...

**Tableaux associatifs (NoSQL)** Structure très simple composée de paires clé → valeur, permettant des recherches très efficaces.

**Relationnelles** MariaDB, PostgreSQL, SQLite, ...



# Systemes d'Information Géographique : QGIS

QGIS 2.14.11-Essen - ProjetTestJoinure

Projet Éditer Vue Couche Préférences Extension Vecteur Raster Base de données Internet Traitement Aide

COMMUNE :: Features total: 10, filtered: 10, selected: 0

ID	DPUL	MULTICAN	J_NOM_COMM	I_X_COMMUNE	I_Y_COMMUNE	J_SUPERFICIE	J_INSEE_CANT	J_INSEE_ARR	J_INSEE_DEPT	J_INSEE_REG	hic_NB_PIETOI	ehic_NB_VELO	hic_NB_VOITUF
0	433	Non	BOUSSE	470515	6745247	1202	17	1	72	52	8793	8000	2000
1	466	Non	LIGRON	474237	6745574	1348	17	1	72	52	6547	6000	500
2	1170	Non	LE BAILLEUL	462145	6746131	2746	17	1	72	52	3589	3000	700
3	807	Non	CRE	464444	6733839	1719	14	1	72	52	3247	3000	600
4	1186	Non	BAZOUGES-...	461769	6736584	2990	14	1	72	52	5679	5000	2400
5	977	Non	VILLAINES-S...	467557	6744178	1916	17	1	72	52	3450	3000	1000
6	922	Non	CROSMIERES	463343	6741281	2045	14	1	72	52	3478	3000	1500
7	606	Non	MAREIL-SUR...	475371	6739051	1183	14	1	72	52	5432	5000	900
8	1186	Non	CLERMONT-...	473148	6741278	1782	14	1	72	52	2345	2000	400
9	15359	Non	LA FLECHE	470872	6737445	7421	14	1	72	52	3761	3000	1200

Montrer toutes les entités

Couches

ATTRIBUTS\_COMMUNES  
Vehicules Feuille1 None

- COMMUNE
- Vehic\_NB\_PIETONS
- Vehic\_NB\_VELOS
- Vehic\_NB\_VOITURES

Bascule le mode d'édition de la couche courante

Coordonnée 457830,6731770 Echelle 1:104779 Rotation 0.0 Rendu EPSG:2154 (OTF)

# Table des matières

## 3 DF Algo. et programmation

- Introduction
- Langages
- Programmation

1 DF Systèmes numériques

2 DF Informations et données

3 DF Algo. et programmation

4 DF Informatique et société

# Binaire - Source

## Binaire

Un ordinateur ne comprend que les zéros et uns dont est composé un fichier binaire.

```
0100010 1001010 01010001  
10010001010000101
```

## Source

Un programme s'écrit dans un langage compréhensible par l'homme et est traduit par un compilateur en binaire.

```
if (argent = bonheur)  
then  
(1/3 population = triste)
```

Exemple de l'utilité de l'ouverture du code source : [Projet télescopique](#)

# Everything as code

## Biologie - Informatique

Tout comportement du vivant peut-il être traité comme le résultat d'un code complexe ? <sup>a</sup>

---

a. Voir le film IROBOT.

## Hardware - software

Qu'est-ce que le software par opposition à l'hardware ?

# Langages de programmation

## Scratch, codeblock



## Javascript, Python, C++, ...

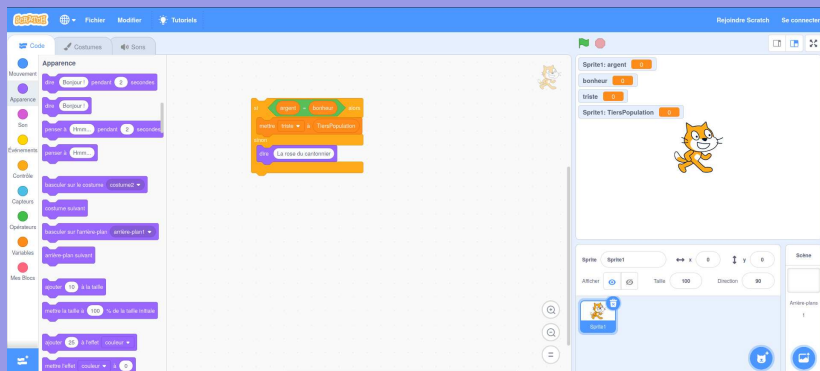
```
if (argent == bonheur)
{1/3 population = triste}
else
{La rose du cantonnier}
```

```
if argent == bonheur :
    1/3 population = triste
else :
    La rose du cantonnier
```

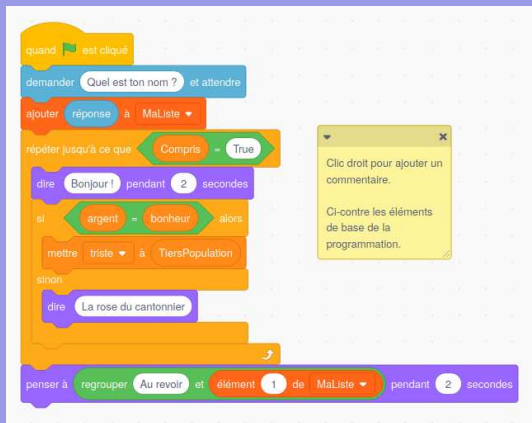
Scratch : <https://scratch.mit.edu/>

L'éditeur de code se trouve à l'adresse :

<https://scratch.mit.edu/projects/editor/>



# Éléments de base de la programmation



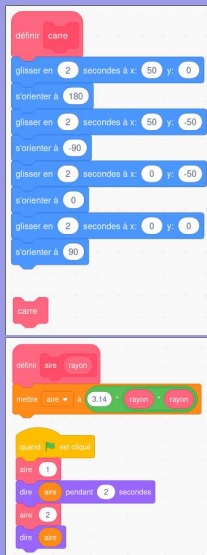
The image shows a Scratch script and a comment box. The script starts with a 'when clicked' event block, followed by a 'ask and wait' block with the text 'Quel est ton nom ?'. Then, it adds the response to a variable named 'MaListe'. A 'repeat until' loop follows, with the condition 'Compris = True'. Inside the loop, there is a 'say' block with the text 'Bonjour !' for 2 seconds. A 'conditional' block follows: 'if argent = bonheur then', which sets 'triste' to 'TiersPopulation'. The 'else' block says 'La rose du cantonnier'. After the loop, there is a 'wait' block with the text 'Au revoir' and 'element 1 of MaListe' for 2 seconds.

Clc droit pour ajouter un commentaire.

Ci-contre les éléments de base de la programmation.

- Entrée, sortie
- Variable ; liste
- Condition logique
- Test, boucle
- Événement
- Commentaires
- Procédure, fonction (voir suite)

# Procédures et fonctions

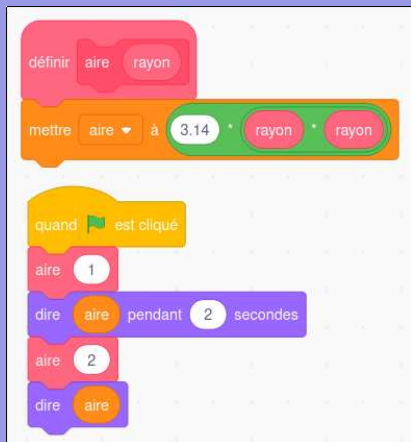


**Procédure** Un code nommé qu'on peut appeler à plusieurs reprises et qui ne retourne rien.

**Fonction** Un code nommé qu'on peut appeler à plusieurs reprises et qui retourne quelque chose.



# Arguments



Un argument est un élément qu'on transmet à une procédure ou une fonction, comme dans le cas ci-contre.

L'argument transmis à la fonction est ici le rayon permettant de calculer l'aire d'un cercle.

Il peut y avoir plusieurs arguments, de différentes natures, comme des nombres, du texte, ...

# Algorithmique et langages

## ALGORITHMIQUE ET PROGRAMMATION

### 1 Instructions élémentaires

#### ALGORITHMIQUE

#### SCRATCH

#### PYTHON

#### 1.1 Affecter une valeur à une variable de type numérique (entier ou flottant)

<pre>compteur ← 3 duree ← 4.5 a ← 2 * (3 + 5) resultat ← 1 + 3a</pre>		<pre>compteur = 3 duree = 4.5 a = 2 * (3 + 5) resultat = 1 + 3 * a</pre>
---	--	--

#### 1.2 Affecter un texte à une variable de type chaîne de caractères

<pre>message ← "Bonjour"</pre>		<pre>message = "Bonjour"</pre>
--------------------------------	--	--------------------------------

#### 1.3 Utiliser une instruction conditionnelle

<pre>Si compteur &gt; 20 alors score ← score + 1 [finSi]</pre>		<pre>if compteur &gt; 20 :     score = score + 1</pre>
<pre>Si compteur &gt; 20 alors score ← score + 1 sinon score ← score - 1 [finSi]</pre>		<pre>if compteur &gt; 20 :     score = score + 1 else :     score = score - 1</pre>

#### 1.4 Utiliser une boucle bornée (nombre d'itérations fixé)

<pre>Répéter 10 fois saut ← saut * 0.95 [finRépéter]</pre>		<pre>for i in range (10) :     saut = saut * 0.95</pre>
<pre>Pour compteur allant de 1 à 9 caval ← caval + compteur [finPour]</pre>		<pre>for compteur in range (1,10) :     caval = caval + compteur</pre>

#### 1.5 Utiliser une boucle non bornée

<pre>Tant que saut &gt;= 1 saut ← saut * 0.95 [finTantQue]</pre>		<pre>while saut &gt;= 1 :     saut = saut * 0.95</pre>
--	--	--

1

Sur la figure ci-contre se trouvent :

**L'algorithme** Un énoncé en mots de la solution d'un problème.

**Scratch** Une présentation dans un langage de blocs.

**Python** Une solution sous forme de code.

La suite de la comparaison : [https://www.carnets.info/formation/tableau\\_AlgoScratch3Python.pdf](https://www.carnets.info/formation/tableau_AlgoScratch3Python.pdf) en remerciant son auteur qui a rendu libre sa documentation.

# Projet SCRATCH

## Au travail

### Votre projet SCRATCH

- Par groupe de deux (même note)
- Avalisé (par votre professeur)
- Bien documenté (la doc. est aussi importante que le code!)

# Table des matières

## 4 DF Informatique et société

1 DF Systèmes numériques

2 DF Informations et données

3 DF Algo. et programmation

4 DF Informatique et société

# L'informatique est-elle un progrès ?

## La technique est-elle vraiment neutre ?

**Consommation d'énergie vs utilité** Dix pourcent de la consommation électrique mondiale (2018), soit quatre pourcent des émissions à effet de serre, avec cinq pourcent d'augmentation tous les ans<sup>a</sup>.

**Sécurité vs vie privée** Rançons, cyberharcèlement, ...

**Traçage et informations personnelles** Internet est le lieu d'un espionnage systématique. Les logiciels des smartphones sont bourrés de traceurs.

**Droit et libertés** Respect du droit sur internet, en particulier respect du droit d'auteur, licences libres.

**Information et secrets** Droit à l'information et droit à l'oubli.

---

a. Voir CNRS : <https://lejournale.cnrs.fr/articles/numerique-le-grand-gachis-energetique>

# L'informatique est-elle un progrès ?

## La technique est-elle vraiment neutre ?

**Robotique** Film Irobot : les robots sont-ils nos amis ? Anecdote des chocolats ...

**Intelligence artificielle** Sélection intelligente des meilleurs enseignants ...

**Big data** Apple et son intelligence raciste ...

**Outils de suivi : reconnaissance faciale, géopositionnement** Être toujours sur ses gardes ...

**Outils de propagande** Fakes news, cambridge analytica (Facebook), résultats américains du moteur de recherche de Google ...

# L'informatique est-elle un progrès ?

## La technique est-elle vraiment neutre ?

**Exploitation humaine** « La machine est ton seigneur et maître  
Analyses, [...] témoignages sur la vie des ouvriers des usines de Foxconn qui la perdent à fabriquer iPhone, [...] » (Yang, Jenny Chan, Xu Lizhi)

**Transports** L'immense majorité des déplacements en voiture se fait sur moins de cinq kilomètres et ne déplace qu'une seule personne. L'immense majorité des messages électroniques ne sert à rien.

**Santé** Opérer à distance. Un bon choix de connaissances ?

**Machinisme** Des ordinateurs surpuissants pour faire quoi ? Des tronçonneuses pour couper les branchettes des haies ?

**Histoire ludique** « Techno-critiques Du refus des machines à la contestation des technosciences » François Jarrige

# Un nuage de gouttes d'énergie : fleuve sans raisons ?

Voiture : contenant deux tonnes pour 80 kg d'humain ; distance moyenne journalière env. 5 km (pour les croissants :-); inefficacité énergétique, pollution sonore, pollution atmosphérique, infrastructure pétrolière polluante ...

## informatique

Smartphones : terres rares, infrastructure d'extraction et de livraison importante, exploitation de travailleurs<sup>a</sup> consommation énergétique très importante (24h/24 pour une utilisation faible; qui éteint son smartphone?), infrastructure de communication faite de routeurs incroyablement nombreux, smog électromagnétique, ... pour « Bon, t'es où ? », « Juste là, je te vois ».

---

a. Voir : « La machine est ton seigneur et ton maître », Yang, Jenny Chan, Xu Lizhi, nov. 2015, Agone



# Vie privée (Affaire Pegasus (2021))

## Affaire des fiches (1989)

« En novembre, cette commission d'enquête annonce ses résultats. [...] Elle critique également la prévention de la Police fédérale contre les contestataires usant de moyens démocratiques et ses méthodes de récolte d'information et annonce qu'elle a découvert que le Ministère public de la Confédération était en possession de fiches contenant des rapports d'observations effectuées sur des personnes et des groupes particuliers en Suisse. Ce qui deviendra « le scandale des fiches » ... » Wikipedia : [https://fr.wikipedia.org/wiki/Scandale\\_des\\_fiches](https://fr.wikipedia.org/wiki/Scandale_des_fiches)

## Edward Snowden (2013)

« Snowden rend publiques, par l'intermédiaire des médias, des informations classées top-sécètes de la NSA concernant la captation des métadonnées des appels téléphoniques aux États-Unis, ainsi que les systèmes d'écoute sur Internet des programmes de surveillance PRISM, XKeyscore, Boundless Informant et Bullrun du gouvernement américain et les programmes de surveillance Tempora, Muscular et Optic Nerve du gouvernement britannique. » Voir : wikipedia : [https://fr.wikipedia.org/wiki/Edward\\_Snowden](https://fr.wikipedia.org/wiki/Edward_Snowden)



21

## Au travail

- 1 Ouvrez Firefox.
- 2 Désactivez les plugins.
- 3 Ouvrez le débogueur (F12).
- 4 Sélectionnez l'onglet réseau.
- 5 Naviguez sur <https://www.laposte.net>
- 6 Observez l'activité réseau pendant une minute.
- 7 Qu'en déduisez-vous ?

# OPTION COMPLÉMENTAIRE

# Table des matières

## 5 OC Notions complémentaires

- Introduction
- Le texte
- L'image
- Le son
- Programmation

1 DF Systèmes numériques

2 DF Informations et données

3 DF Algo. et programmation

4 DF Informatique et société

# Évaluations

Désagréables  $\longleftrightarrow$  agréables<sup>22</sup> ?



Description des évaluations

# Traitements de texte

Objectifs : mettre en forme du texte. Moyens dépendant du contexte :

- **Traitements de texte Wysiwyg** (What You See Is What You Get) (Word<sup>®</sup>, LibreOffice, Scribus, ...) : soulignés, gras, italique, etc.

*Affiches et petits documents.*

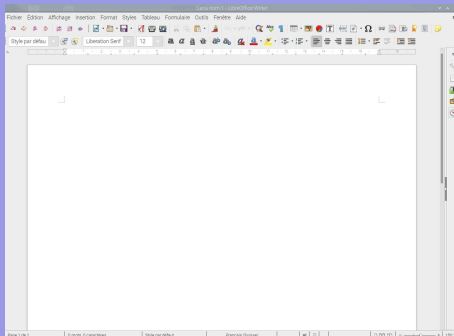
- **Traitements de texte balisés** (xhtml, latex, XML, ...) :  
`<p align='center'> texte </p>`,  
`\begin{center} texte \end{center}`, etc.

*Internet, long documents, documentation.*



# LibreOffice

Traitement de texte classique. Bien que ...



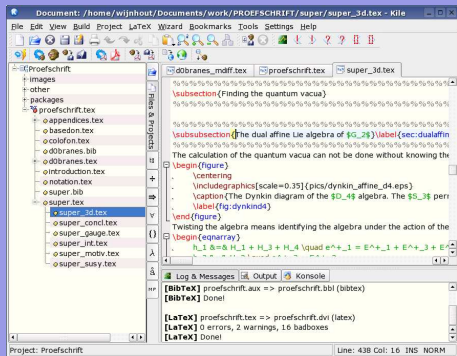
Gestion avancée des styles, références, bibliographie, index, ...

LibreOffice avec styles



# Latex

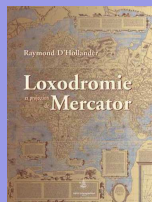
Traitement de texte haute qualité :  
gestion professionnelle automatique de la typographie.



Exemples :

*"L'astrolabe, histoire théorie et pratique"* et

*"Loxodromie et projection de Mercator"*.

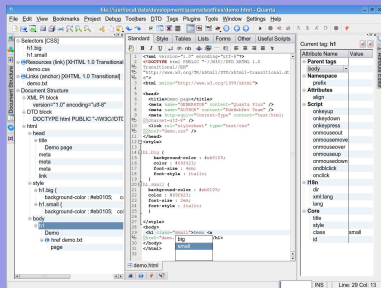
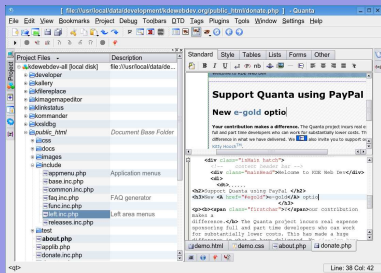


Autre wijnexample connu : votre "formulaire et tables"...

# Xhtml et Css

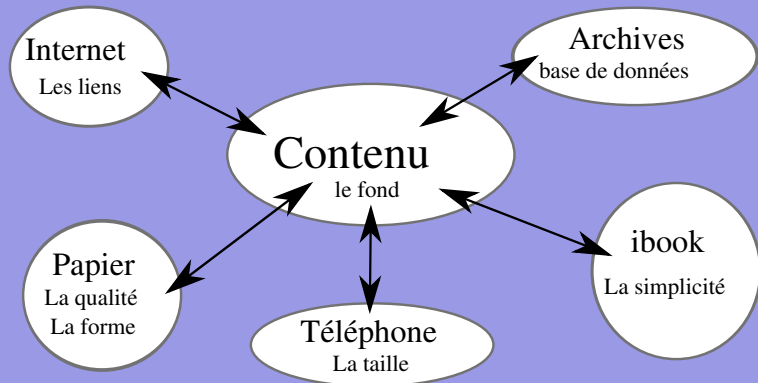
Orientation web – séparation du contenu et de la forme.

Langages balisés sans boucles, fortement normalisés et strict.



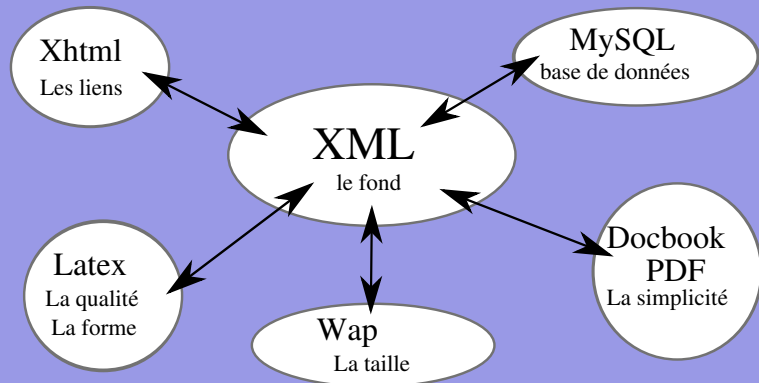
# Publication multi-support

Aujourd'hui les supports d'édition sont multiples.



# Publication multi-support

Et les langages nécessaires aussi ( $\text{\LaTeX}$ et MySQL ne sont pas XML).



# XML (eXtensible Markup Language)

Idée : structurer du contenu à l'aide de balises : `<` et `>`.

Exemple : `<title>`Le XML (eXtensible Markup Language)`</title>`.

Le XML n'impose aucun contenu aux balises `<` `>`.

Chaque langage va définir lui-même ses propres balises.

Ainsi, en Xhtml, on trouve la balise "title" qui peut ne pas apparaître dans un autre langage XML.

# DocBook

Un autre exemple de XML est donné par DocBook dont voici un exemple :

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE article PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN">
<article lang=""> <para>PROPOSITION XXV - PROBLEME VI</para>
<para>Trouver les forces du Soleil.</para> <para>
<inlinegraphic fileref="embedded:Image1" width="12.78cm" depth="4.922cm"/>
Que S représente le Soleil, T la Terre, P la Lune</para>
<para>FIG 3.</para> </article>
```

# WAP

Un autre exemple de XML est donné par le WAP (Wireless Application Protocol).

Le langage utilisé est le WML (Wireless Markup Language).

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml"> <wml>
<card id="cours" title="||cours de physique||"> <p align="left">
Ma première <br/>page WML </p> </card> </wml>
```

# Nombres à virgule flottante

L'écriture d'un nombre est facilitée si elle se fait sous la forme :

$$\text{nombre} = (-1)^{\text{signe}} \cdot \text{mantisse} \cdot \text{base}^{\text{exposant}}$$

avec  $1 \leq \text{mantisse} < 10$

Par exemple le nombre décimal  $-234$  peut s'écrire :  $-2,34 \cdot 10^2$  où le signe vaut 1, la mantisse 2,34, la base 10 et l'exposant 2.

Un nombre binaire peut être représenté de la même manière en base deux à l'aide du signe 0 ou 1, d'une mantisse et d'un exposant.



# Représentation d'un nombre sur 32 bits

La représentation d'un nombre binaire à virgule flottante :

$$\text{nombre} = (-1)^{\text{signe}} \cdot \text{mantisse} \cdot \text{base}^{\text{exposant}}$$

avec  $1 \leq \text{mantisse} < 2$

est déterminée par le nombre de bits qui lui sont associés.

Couramment on utilise 32 bits répartis ainsi :

**Le signe** 1 bit, 0 (positif) ou 1 (négatif).

**L'exposant** 8 bits, soit 256 nombres. Biaisé de 127 pour permettre des exposants négatifs, soit de  $-127$  à  $+127$ .

**La mantisse** 23 bits, sans compter le 1 précédant toujours la virgule.

# Exemple numérique

$$\begin{aligned} \text{Nombre} &= 101010000000000000000000000000110 \\ &= 1 \mid 00000110 \mid 010100000000000000000000 \\ &= \text{signe} \mid \text{exposant} \mid \text{mantisse} \end{aligned}$$

Ce nombre peut être traduit en décimal de la manière suivante :

- Le signe = 1  $\Rightarrow$  nombre négatif.
- L'exposant vaut :  $0 \cdot 2^0 + 1 \cdot 2^1 + 1 \cdot 2^2 = 6$ , c'est-à-dire en réalité  $6 - 127 = -121$  à cause du biais.
- La mantisse =  $1 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} + 0 \cdot 2^{-3} + 1 \cdot 2^{-4} = 1,3125$

Soit le nombre :  $-1,3125 \cdot 2^{-121} = -4,937 \cdot 10^{-37}$

# Exercices

Convertissez les nombres suivants :

- 01000000001100000000000000000000 en décimal
- 01100000001110000000000000000000 en décimal
- 01010000001010000000000000000000 en décimal
- 10101000001100000000000000000000 en décimal

# Exercices

Convertissez les nombres suivants :

- 0 | 10000000 | 011000000000000000000000
- 01100000001110000000000000000000 en décimal
- 01010000001010000000000000000000 en décimal
- 10101000001100000000000000000000 en décimal

# Exercices

Convertissez les nombres suivants :

- $0 \mid 10000000 \mid 011000000000000000000000$   
 $+ \mid 2^7 - 127 = 1 \mid 1 + 2^{-2} + 2^{-3} = 1,375 \Rightarrow$
- $01100000001110000000000000000000$  en décimal
- $01010000001010000000000000000000$  en décimal
- $10101000001100000000000000000000$  en décimal

# Exercices

Convertissez les nombres suivants :

- $0 \mid 10000000 \mid 011000000000000000000000$   
 $+ \mid 2^7 - 127 = 1 \mid 1 + 2^{-2} + 2^{-3} = 1,375 \Rightarrow$   
 $+1,375 \cdot 2^1 = 2,75$
- $01100000001110000000000000000000$  en décimal
- $01010000001010000000000000000000$  en décimal
- $10101000001100000000000000000000$  en décimal

# Exercices

Convertissez les nombres suivants :

- $01000000001100000000000000000000=$   
 $+1,375 \cdot 2^1 = 2,75$
- $01100000001110000000000000000000$  en décimal
- $01010000001010000000000000000000$  en décimal
- $10101000001100000000000000000000$  en décimal

# Exercices

Convertissez les nombres suivants :

- $01000000001100000000000000000000=$   
 $+1,375 \cdot 2^1 = 2,75$
- $0 \mid 1100000 \mid 0111000000000000000000$
- $01010000001010000000000000000000$  en décimal
- $10101000001100000000000000000000$  en décimal



# Exercices

Convertissez les nombres suivants :

- $01000000001100000000000000000000=$   
 $+1,375 \cdot 2^1 = 2,75$
- $0 \mid 11000000 \mid 011100000000000000000000$   
 $+ \mid 2^7 + 2^6 - 127 = 65 \mid 1 + 2^{-2} + 2^{-3} + 2^{-4} = 1,4375 \Rightarrow$
- $01010000001010000000000000000000$  en décimal
- $10101000001100000000000000000000$  en décimal

# Exercices

Convertissez les nombres suivants :

- $01000000001100000000000000000000=$   
 $+1,375 \cdot 2^1 = 2,75$
- $0 \mid 11000000 \mid 011100000000000000000000$   
 $+ \mid 2^7 + 2^6 - 127 = 65 \mid 1 + 2^{-2} + 2^{-3} + 2^{-4} = 1,4375 \Rightarrow$   
 $+1,4375 \cdot 2^{65} = 5,3 \cdot 10^{19}$
- $01010000001010000000000000000000$  en décimal
- $10101000001100000000000000000000$  en décimal

# Exercices

Convertissez les nombres suivants :

- $01000000001100000000000000000000=$   
 $+1,375 \cdot 2^1 = 2,75$
- $01100000001110000000000000000000=$   
 $+1,4375 \cdot 2^{65} = 5,3 \cdot 10^{19}$
- $01010000001010000000000000000000$  en décimal
- $10101000001100000000000000000000$  en décimal

# Exercices

Convertissez les nombres suivants :

- $01000000001100000000000000000000=$   
 $+1,375 \cdot 2^1 = 2,75$
- $01100000001110000000000000000000=$   
 $+1,4375 \cdot 2^{65} = 5,3 \cdot 10^{19}$
- $0 \mid 10100000 \mid 010100000000000000000000$
- $10101000001100000000000000000000$  en décimal

# Exercices

Convertissez les nombres suivants :

- $01000000001100000000000000000000=$   
 $+1,375 \cdot 2^1 = 2,75$
- $01100000001110000000000000000000=$   
 $+1,4375 \cdot 2^{65} = 5,3 \cdot 10^{19}$
- $0 \mid 10100000 \mid 010100000000000000000000$   
 $+ \mid 2^7 + 2^5 - 127 = 33 \mid 1 + 2^{-2} + 2^{-4} = 1,3125 \Rightarrow$
- $10101000001100000000000000000000$  en décimal

# Exercices

Convertissez les nombres suivants :

- $01000000001100000000000000000000=$   
 $+1,375 \cdot 2^1 = 2,75$
- $01100000001110000000000000000000=$   
 $+1,4375 \cdot 2^{65} = 5,3 \cdot 10^{19}$
- $0 \mid 10100000 \mid 010100000000000000000000$   
 $+ \mid 2^7 + 2^5 - 127 = 33 \mid 1 + 2^{-2} + 2^{-4} = 1,3125 \Rightarrow$   
 $+1,3125 \cdot 2^{33} = 1,1 \cdot 10^{10}$
- $10101000001100000000000000000000$  en décimal

# Exercices

Convertissez les nombres suivants :

- $01000000001100000000000000000000=$   
 $+1,375 \cdot 2^1 = 2,75$
- $01100000001110000000000000000000=$   
 $+1,4375 \cdot 2^{65} = 5,3 \cdot 10^{19}$
- $01010000001010000000000000000000=$   
 $+1,3125 \cdot 2^{33} = 1,1 \cdot 10^{10}$
- $10101000001100000000000000000000$  en décimal

# Exercices

Convertissez les nombres suivants :

- $01000000001100000000000000000000=$   
 $+1,375 \cdot 2^1 = 2,75$
- $01100000001110000000000000000000=$   
 $+1,4375 \cdot 2^{65} = 5,3 \cdot 10^{19}$
- $01010000001010000000000000000000=$   
 $+1,3125 \cdot 2^{33} = 1,1 \cdot 10^{10}$
- $1 \mid 01010000 \mid 011000000000000000000000$



# Exercices

Convertissez les nombres suivants :

- $01000000001100000000000000000000=$   
 $+1,375 \cdot 2^1 = 2,75$
- $01100000001110000000000000000000=$   
 $+1,4375 \cdot 2^{65} = 5,3 \cdot 10^{19}$
- $01010000001010000000000000000000=$   
 $+1,3125 \cdot 2^{33} = 1,1 \cdot 10^{10}$
- $1 \mid 01010000 \mid 011000000000000000000000$   
 $- \mid 2^6 + 2^4 - 127 = -47 \mid 1 + 2^{-2} + 2^{-3} = 1,375 \Rightarrow$

# Exercices

Convertissez les nombres suivants :

- $01000000001100000000000000000000=$   
 $+1,375 \cdot 2^1 = 2,75$
- $01100000001110000000000000000000=$   
 $+1,4375 \cdot 2^{65} = 5,3 \cdot 10^{19}$
- $01010000001010000000000000000000=$   
 $+1,3125 \cdot 2^{33} = 1,1 \cdot 10^{10}$
- $1 \mid 01010000 \mid 011000000000000000000000$   
 $- \mid 2^6 + 2^4 - 127 = -47 \mid 1 + 2^{-2} + 2^{-3} = 1,375 \Rightarrow$   
 $-1,375 \cdot 2^{-47} = -9,8 \cdot 10^{-15}$

# Exercices

Convertissez les nombres suivants :

- $01000000001100000000000000000000=$   
 $+1,375 \cdot 2^1 = 2,75$
- $01100000001110000000000000000000=$   
 $+1,4375 \cdot 2^{65} = 5,3 \cdot 10^{19}$
- $01010000001010000000000000000000=$   
 $+1,3125 \cdot 2^{33} = 1,1 \cdot 10^{10}$
- $10101000001100000000000000000000=$   
 $-1,375 \cdot 2^{-47} = -9,8 \cdot 10^{-15}$

# Byte, bit et octet

## Décimal

0, 1, 2, ..., 9 : chiffres.

23, 45, 234, ... : nombres.

# Byte, bit et octet

## Décimal

0, 1, 2, ..., 9 : chiffres.

23, 45, 234, ... : nombres.

## Binaire

0 et 1 : **bits**.

101, 11, 10, ... : nombres.

# Byte, bit et octet

## Décimal

0, 1, 2, ..., 9 : chiffres.

23, 45, 234, ... : nombres.

## Binaire

0 et 1 : **bits**.

101, 11, 10, ... : nombres.

(Attention, 101 binaire ne se lit pas "cent un" et ne correspond pas à la valeur décimale 101. En réalité, on a :

$0_b = 0_d$  ;  $1_b = 1_d$  ;  $10_b = 2_d$  ;  $11_b = 3_d$  ;  $100_b = 4_d$  ;  $101_b = 5_d$  ...)

# Byte, bit et octet

- Avec un *bit*, on a les nombres 0 et 1, soit deux nombres.

# Byte, bit et octet

- Avec un *bit*, on a les nombres 0 et 1, soit deux nombres.
- Avec deux *bits*, on a les nombres 00, 01, 10 et 11, soit quatre nombres.



# Byte, bit et octet

- Avec un *bit*, on a les nombres 0 et 1, soit deux nombres.
- Avec deux *bits*, on a les nombres 00, 01, 10 et 11, soit quatre nombres.
- Avec trois *bits*, on a les nombres 000, 001, 010, 011, 100, 101, 110 et 111, soit huit nombres.

# Byte, bit et octet

- Avec un *bit*, on a les nombres 0 et 1, soit deux nombres.
- Avec deux *bits*, on a les nombres 00, 01, 10 et 11, soit quatre nombres.
- Avec trois *bits*, on a les nombres 000, 001, 010, 011, 100, 101, 110 et 111, soit huit nombres.
- Avec quatre *bits*, on a ... ?

# Byte, bit et octet

- Avec un *bit*, on a les nombres 0 et 1, soit deux nombres.
- Avec deux *bits*, on a les nombres 00, 01, 10 et 11, soit quatre nombres.
- Avec trois *bits*, on a les nombres 000, 001, 010, 011, 100, 101, 110 et 111, soit huit nombres.
- Avec quatre *bits*, on a ... ?
- Avec  $n$  *bits*, on a un **arrangement avec répétition de  $k$  éléments choisis parmi deux** (0 ou 1). Les mathématiques donnent alors le nombre de ces arrangements :

$$\bar{A}_k^2 = 2^k$$

# Byte, bit et octet

Avec huit bits, on a donc :

$$\bar{A}_8^2 = 2^8 = 256 \text{ possibilités}$$

On peut donc représenter 256 nombres.

Un **octet** est un nombre de huit bits.  
Un *octet* est aussi nommé **byte**.

# kilobyte, Megabyte, Gigabyte, ...

## Décimal

kilo  $\equiv$  nombre de possibilités de réaliser un nombre à l'aide de 3 chiffres :

$$\bar{A}_3^{10} = 10^3 = 1000 \text{ possibilités}$$

# kilobyte, Megabyte, Gigabyte, ...

## Décimal

kilo  $\equiv$  nombre de possibilités de réaliser un nombre à l'aide de 3 chiffres :

$$\bar{A}_3^{10} = 10^3 = 1000 \text{ possibilités}$$

## Binaire

kilo  $\equiv$  nombre de possibilités de réaliser un nombre à l'aide de 10 bits :

$$\bar{A}_{10}^2 = 2^{10} = 1024 \text{ possibilités}$$

# kilobyte, Megabyte, Gigabyte, ...

- 1 Un kilobyte ou kilooctet, noté ko, correspond à 1024 octets.
- 2 Un Megabyte ou Megaoctet, noté Mo, correspond à 1024 kilooctet, soit 1'048'576 bytes ou octets.
- 3 Un Gigabyte ou Gigaoctet, noté Go, correspond à 1024 Megaoctet, soit 1'048'576 kilobyte ou 1'073'741'824 bytes.

L'ordre de grandeur correspond à la notation décimale :

- 1 kilo pour milliers.
- 2 Mega pour millions.
- 3 Giga pour milliards.

# Exercices

Sur un processeur 16 bits, combien de nombres entiers non signés peut on représenter ?

(un processeur 16 bits peut manipuler des “mots” de 16 bits, exemple 1110110000110110)



# Exercices

Sur un processeur 16 bits, combien de nombres entiers non signés peut on représenter ?

(un processeur 16 bits peut manipuler des “mots” de 16 bits, exemple 1110110000110110)

On a  $2^{16} = 65'536$  possibilités, soit autant de nombres.

# Exercices

Sur le même processeur, quels sont les plus petits et plus grands nombres entiers signés qui puissent être représentés ?  
(c'est à dire avec la possibilité de faire des nombres positifs et négatifs)

# Exercices

Sur le même processeur, quels sont les plus petits et plus grands nombres entiers signés qui puissent être représentés ?

(c'est à dire avec la possibilité de faire des nombres positifs et négatifs)

On a un bit de signe (0 ou 1) et 15 bits pour le nombre, soit  $2^{15} = 32'768$  possibilités. On peut donc représenter des nombres de -32'768 à 32'767 (à cause du zéro).

# Exercices

Sur un processeur 32 bits, si on met dans un fichier un fois tous les nombres entiers non signés, quel sera en Go la taille du fichier ?

# Exercices

Sur un processeur 32 bits, si on met dans un fichier un fois tous les nombres entiers non signés, quel sera en Go la taille du fichier ?

Avec un tel processeur on a  $2^{32} = 4'294'967'296$  nombres possibles (avec le zéro). Chaque nombre est codé sur 32 bits, soit au total  $4'294'967'296 \cdot 32 = 1,37 \cdot 10^{11}$  bits. Cela représente  $1,71 \cdot 10^{10}$  octets ou 16'777'216 ko, ou encore 16'384 Mo, c'est-à-dire 16 Go.

# Exercices

Quelle plage de nombres réels couvre la représentation d'un réel (1 bit pour le signe, 23 bits pour la mantisse et 8 bits pour l'exposant).

# Exercices

Quelle plage de nombres réels couvre la représentation d'un réel (1 bit pour le signe, 23 bits pour la mantisse et 8 bits pour l'exposant).

Le nombre maximum codé par la mantisse est

$$1 + \sum_{i=-1}^{-23} 2^i = 1,99999988079071 \cong 2$$

Comme l'exposant est codé sur 8 bits, on a  $2^8 = 256$  possibilités. Pour coder les négatifs on biaise de 127. Cela signifie que le nombre maximum codé par l'exposant est

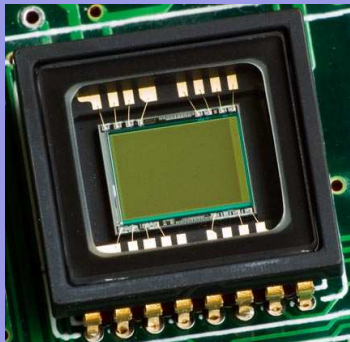
$$2^{127} = 1,7 \cdot 10^{38}$$

Au total, on a donc le nombre maximal

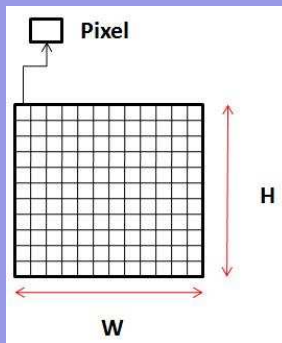
$$1,9999 \cdot 1,7 \cdot 10^{38} = 3,4 \cdot 10^{38}$$

# Image matricielle ou bitmap

Capteur ccd<sup>23</sup> : niveaux de gris.



Grille de pixels<sup>24</sup>



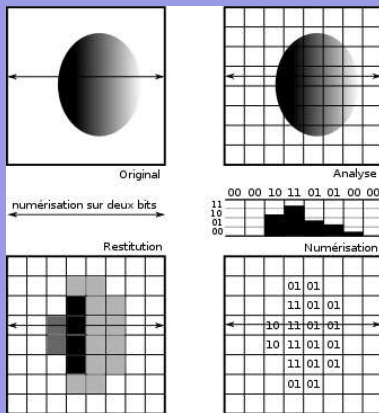


# Numérisation sur deux bits

Sur deux bits, on a quatre niveaux de gris possibles :

- 00 blanc,
- 01 gris clair,
- 10 gris foncé et
- 11 noir.

Sur huit bits, on aurait 256 niveaux de gris, mais la forme de l'image resterait identique car elle dépend de la résolution.



# Résolution



Unités de la résolution<sup>25</sup> : dpi (dot per inch) ou ppp (point par pouce).

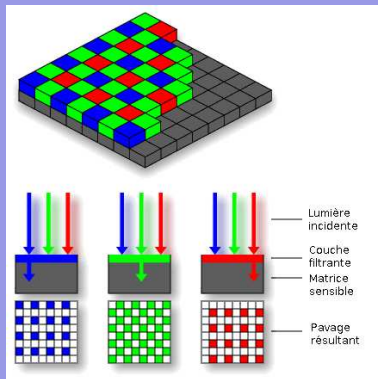
Exemples typiques : écran 75-100 dpi, photocopieuse 360 dpi, imprimante 700 dpi, scanner 2000 dpi.

# Image en couleur : codage RVB.

**Idée :** masquer les photosites par une couche mince d'oxyde de silicium filtrant les trois couleurs de base : rouge (R), vert (V) et bleu (B).

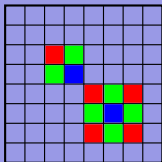
**Norme :** par la Commission Internationale de l'éclairage aux longueurs d'ondes de 546,1 nanomètre pour le rouge, 700,0 nanomètre pour le vert et 435,8 nanomètre pour le bleu.

## Filtre de Bayer<sup>26</sup>.



# Filtre de Bayer

## Filtre de Bayer.

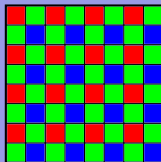


Motif de base

$$R = (237 + 63 + 255 + 150) / 4 = 176$$

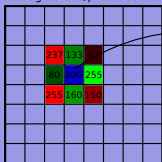
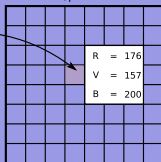
$$V = (133 + 80 + 255 + 160) / 4 = 157$$

$$B = 200$$



Pavage

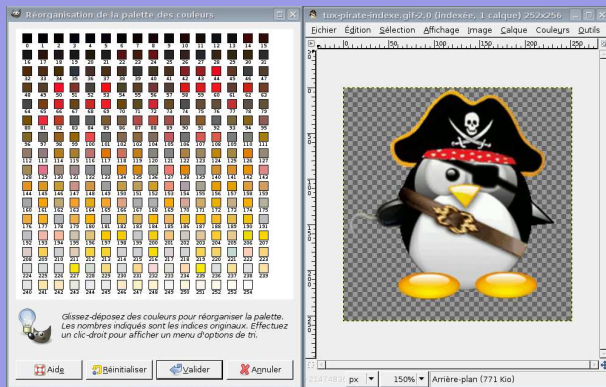
Codage 8 bits/pixel décimal

Décodage pixel central  
24 bits/pixel décimal

- Pavage privilégiant le vert (œil plus sensible),
- Couleur pixel central :  
Bleu = 200  
Rouge =  $(237 + 63 + 255 + 150) / 4 = 176$   
Vert =  $(133 + 80 + 255 + 160) / 4 = 157$
- Codage en 24 bits : 8 bits / couleur.  
Soit 256 niveaux de luminosité pour chaque couleur, c'est-à-dire  $256 \cdot 256 \cdot 256 = 16'777'216$  couleurs différentes. La palette comporte plus de 16 millions de couleurs.

# Palette de couleur

Autres manière de coder les images : la palette de couleur.



- Choix : **256** couleurs
- Indexation de chaque pixel par l'une de celle-ci.

# Taille et poids d'une image

Couleurs vraies : 24 bits / pixel

Image  $252 \times 256$  pixels =  
64'512 pixels

Couleurs vraies : 3 octets / pixel  
 $3 \cdot 64'512 = 193'536$  octets

$193'536$  octets = 189 ko

Total : **189 ko**

En réalité : 65,2 ko (compression)

# Taille et poids d'une image

## Couleurs vraies : 24 bits / pixel

Image  $252 \times 256$  pixels =  
64'512 pixels

Couleurs vraies : 3 octets / pixel  
 $3 \cdot 64'512 = 193'536$  octets

$193'536$  octets = 189 ko

Total : **189 ko**

En réalité : 65,2 ko (compression)

## Couleurs indexées

Image  $252 \times 256$  pixels =  
64'512 pixels

Index sur 8 bits : 1 octet/pixel  
 $64'512$  octets = 63 ko

Palette  $256 \times 24$  bits = 6'144 bits  
 $6'144$  bits = 768 octets = 0,75 ko

Total : **63,75 ko**

En réalité : 18 ko (compression)

# Formats d'image : fichier .bmp (bitmap)

Éditeur  
texte

Éditeur binaire

BMF.....

..6..(.

.....

.....

.....

.....

.....ÿ.

..ÿÿÿ....

ÿ.ÿ...

01000010	01001101	01000110	00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00110110	00000000	00000000	00000000	00000000	00101000	00000000
00000000	00000000	00000010	00000000	00000000	00000000	00000000	00000010	00000000
00000000	00000000	00000001	00000000	00011000	00000000	00000000	00000000	00000000
00000000	00000000	00010000	00000000	00000000	00000000	00010011	00001011	00000000
00000000	00000000	00010011	00001011	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	11111111	00000000	00000000
00000000	11111111	11111111	11111111	00000000	00000000	00000000	00000000	00000000
11111111	00000000	11111111	00000000	00000000	00000000	00000000	00000000	00000000



# Codage hexadécimal (fichier .bmp)

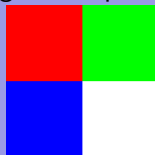
## Hexadécimal

```

42 4d 46 00 00 00 00 00 00 00 36 00 00 00 28 00 BMF.....6... (.
00 00 02 00 00 00 02 00 00 00 01 00 18 00 00 00 .....
00 00 10 00 00 00 13 0b 00 00 13 0b 00 00 00 00 .....
00 00 00 00 00 00 ff 00 00 ff ff ff 00 00 00 00 .....ÿ..ÿÿÿ...
ff 00 ff 00 00 00                               ÿ.ÿ...

```

L'image correspondante :



# Structure du fichier .bmp

## Table – Format d'un bitmap

fichier	42 4D 46 00 00 00 00 00 00 00 36 00 00 00	Caractère B et M pour désigner un bitmap Taille du fichier 46 = 70 octets Réservé toujours à 0 Position (offset) de l'image 36 = 54 octets
Entêtes image	28 00 00 00 02 00 00 00 02 00 00 00 01 00 18 00 00 00 00 00 10 00 00 00 13 0b 00 00 13 0b 00 00 00 00 00 00 00 00 00 00	Taille de l'entête du fichier 28 = 40 octets Largeur de l'image : 2 pixels Hauteur de l'image : 2 pixels nombre de plans utilisés nombre de bits par pixel 18 = 24 bits/pixel (3 octets) Méthode de compression (0 pas de compression) Taille de l'image 10 = 16 octets Résolution horizontale 0b13 = 2835 pixel par mètre Résolution verticale 0b13 = 2835 pixel par mètre Nombre de couleurs de la palette : 0 palette complète Nombre de couleurs importantes de la palette
Image	ff 00 00 ff ff ff 00 00 00 00 ff 00 ff 00 00 00	B=255,V=0,R=0 : bleu B=255, V=255, R=255 : blanc Pour que la ligne ait un nb d'octets multiple de 4 B=0, V=0, R=255 : rouge B=0, V=255, R=0 : vert Pour que la ligne ait un nb d'octets multiple de 4

# Hexadécimal

Un nombre hexadécimal est constitué de 16 chiffres différents. La correspondance avec le décimal est la suivante :

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

A l'instar de la base 10, le correspondant décimal d'un nombre hexadécimal se calcule de la manière suivante :

$$\text{Décimal : } 930 = 9 \cdot 10^2 + 3 \cdot 10^1 + 0 \cdot 10^0$$

$$\begin{aligned}\text{Hexadécimal : } 3A2 &= 3 \cdot 16^2 + A \cdot 16^1 + 2 \cdot 16^0 \\ &= 768 + 10 \cdot 16 + 2 = 930\end{aligned}$$

# Exercices

1→1 ... 9→9 / 10→A / 11→B / 12→C / 13→D / 14→E / 15→F

- Convertissez les nombres hexadécimaux suivants en décimal : 5F, 12, 5, A1, DD, F, 4E, 3B et 43.
- Convertissez les nombres décimaux suivants en hexadécimal : 4, 9, 12, 15, 22, 24, 33, 102, 250, 4343 et 3853.
- En vraie couleur, donnez le code hexadécimal du rouge, du bleu et du vert.
- En vraie couleur, donnez le code hexadécimal de la couleur comprenant 30% de rouge, 20% de vert et 10% de bleu.
- En vraie couleur, donnez le code hexadécimal de la couleur comprenant 50% de rouge, 80% de vert et 3% de bleu.

# Exercices

$1 \rightarrow 1 \dots 9 \rightarrow 9 / 10 \rightarrow A / 11 \rightarrow B / 12 \rightarrow C / 13 \rightarrow D / 14 \rightarrow E / 15 \rightarrow F$

- Convertissez les nombres hexadécimaux suivants en décimal :  
5F, 12, 5, A1, DD, F, 4E, 3B et 43.

$$5F_{hex} =_{dec} 5 \cdot 16 + F = 80 + 15 = 95$$

# Exercices

$1 \rightarrow 1 \dots 9 \rightarrow 9 / 10 \rightarrow A / 11 \rightarrow B / 12 \rightarrow C / 13 \rightarrow D / 14 \rightarrow E / 15 \rightarrow F$

- Convertissez les nombres hexadécimaux suivants en décimal :  
5F, 12, 5, A1, DD, F, 4E, 3B et 43.

$$12_{hex} =_{dec} 1 \cdot 16 + 2 = 18$$

# Exercices

1→1 ... 9→9 / 10→A / 11→B / 12→C / 13→D / 14→E / 15→F

- Convertissez les nombres hexadécimaux suivants en décimal :  
5F, 12, 5, A1, DD, F, 4E, 3B et 43.

$$5_{hex} =_{dec} 5, A1_{hex} =_{dec} 10 \cdot 16 + 1 = 161, DD_{hex} =_{dec} 221, \\ F_{hex} =_{dec} 15, 4E_{hex} =_{dec} 78, 3B_{hex} =_{dec} 59 \text{ et } 43_{hex} =_{dec} 67$$

# Exercices

$1 \rightarrow 1 \dots 9 \rightarrow 9 / 10 \rightarrow A / 11 \rightarrow B / 12 \rightarrow C / 13 \rightarrow D / 14 \rightarrow E / 15 \rightarrow F$

- Convertissez les nombres décimaux suivants en hexadécimal :  
4, 9, 12, 15, 22, 24, 33, 102, 250, 4343 et 3853.



# Exercices

$1 \rightarrow 1 \dots 9 \rightarrow 9 / 10 \rightarrow A / 11 \rightarrow B / 12 \rightarrow C / 13 \rightarrow D / 14 \rightarrow E / 15 \rightarrow F$

- Convertissez les nombres décimaux suivants en hexadécimal :  
4, 9, 12, 15, 22, 24, 33, 102, 250, 4343 et 3853.  
 $4_{dec} =_{hex} 4$ ,  $9_{dec} =_{hex} 9$ ,  $12_{dec} =_{hex} C$ ,  $15_{dec} =_{hex} F$

# Exercices

$1 \rightarrow 1 \dots 9 \rightarrow 9 / 10 \rightarrow A / 11 \rightarrow B / 12 \rightarrow C / 13 \rightarrow D / 14 \rightarrow E / 15 \rightarrow F$

- Convertissez les nombres décimaux suivants en hexadécimal :

4, 9, 12, 15, 22, 24, 33, 102, 250, 4343 et 3853.

$$22_{dec} =_{hex} 1 \cdot 16 + 6 = 16, \quad 24_{dec} =_{hex} 1 \cdot 16 + 8 = 18,$$

$$33_{dec} =_{hex} 2 \cdot 16 + 1 = 21$$

# Exercices

1→1 ... 9→9 / 10→A / 11→B / 12→C / 13→D / 14→E / 15→F

- Convertissez les nombres décimaux suivants en hexadécimal :  
4, 9, 12, 15, 22, 24, 33, 102, 250, 4343 et 3853.

$$102_{dec} =_{hex} 6 \cdot 16 + 6 = 66, 250_{dec} =_{hex} 15 \cdot 16 + 10 = FA$$

# Exercices

$1 \rightarrow 1 \dots 9 \rightarrow 9 / 10 \rightarrow A / 11 \rightarrow B / 12 \rightarrow C / 13 \rightarrow D / 14 \rightarrow E / 15 \rightarrow F$

- Convertissez les nombres décimaux suivants en hexadécimal :  
4, 9, 12, 15, 22, 24, 33, 102, 250, 4343 et 3853.

$$4343_{dec} =_{hex} 1 \cdot 16^3 + 0 \cdot 16^2 + 15 \cdot 16^1 + 7 \cdot 16^0 = 10F7$$

# Exercices

$1 \rightarrow 1 \dots 9 \rightarrow 9 / 10 \rightarrow A / 11 \rightarrow B / 12 \rightarrow C / 13 \rightarrow D / 14 \rightarrow E / 15 \rightarrow F$

- Convertissez les nombres décimaux suivants en hexadécimal :  
4, 9, 12, 15, 22, 24, 33, 102, 250, 4343 et 3853.  
 $3853_{dec} =_{hex} 15 \cdot 16^2 + 0 \cdot 16^1 + 13 \cdot 16^0 = F0D$

# Exercices

1→1 ... 9→9 / 10→A / 11→B / 12→C / 13→D / 14→E / 15→F

- En vraie couleur, donnez le code hexadécimal du rouge, du bleu et du vert.

# Exercices

1→1 ... 9→9 / 10→A / 11→B / 12→C / 13→D / 14→E / 15→F

- En vraie couleur, donnez le code hexadécimal du rouge, du bleu et du vert.

FF0000, pour le rouge car on a 8 bits / couleur, soit 256 niveaux de couleur : deux chiffres hexadécimaux =  $16 \cdot 16 = 256$

0000FF, pour le bleu et 00FF00 pour le vert, car le codage est RVB.

# Exercices

1→1 ... 9→9 / 10→A / 11→B / 12→C / 13→D / 14→E / 15→F

- En vraie couleur, donnez le code hexadécimal de la couleur comprenant 30% de rouge, 20% de vert et 10% de bleu.



# Exercices

$1 \rightarrow 1 \dots 9 \rightarrow 9 / 10 \rightarrow A / 11 \rightarrow B / 12 \rightarrow C / 13 \rightarrow D / 14 \rightarrow E / 15 \rightarrow F$

- En vraie couleur, donnez le code hexadécimal de la couleur comprenant 30% de rouge, 20% de vert et 10% de bleu.  
30% R :  $0,3 \cdot 256 = 77_{dec} =_{hex} 4D$ , 20% V :  $51_{dec} =_{hex} 33$ ,  
10% B :  $26_{dec} =_{hex} 1A$ , soit 4D331A

# Exercices

1→1 ... 9→9 / 10→A / 11→B / 12→C / 13→D / 14→E / 15→F

- En vraie couleur, donnez le code hexadécimal de la couleur comprenant 50% de rouge, 80% de vert et 3% de bleu.

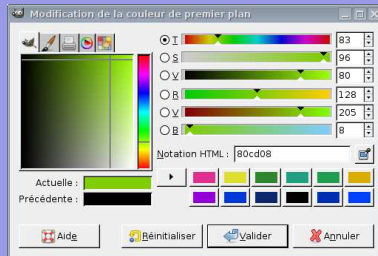
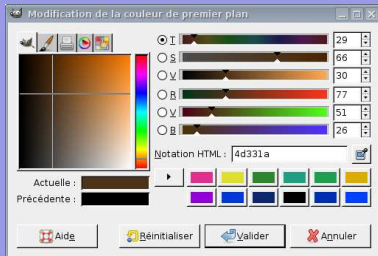
# Exercices

1→1 ... 9→9 / 10→A / 11→B / 12→C / 13→D / 14→E / 15→F

- En vraie couleur, donnez le code hexadécimal de la couleur comprenant 50% de rouge, 80% de vert et 3% de bleu.  
50% R :  $0,5 \cdot 256 = 128_{dec} =_{hex} 80$ , 80% V :  
 $205_{dec} =_{hex} CD$ , 3% B :  $8_{dec} =_{hex} 08$ , soit 80CD08

## Gimp

Les deux couleurs définies dans l'exercice précédent sont :



# Exercice : ceci est une image ...

```
42 4d de 00 00 00 00 00 00 00 00 36 00 00 00 28 00
00 00 07 00 00 00 07 00 00 00 01 00 18 00 00 00
00 00 a8 00 00 00 13 0b 00 00 13 0b 00 00 00 00
00 00 00 00 00 00 ff ff ff ff ff ff ff ff ff 00
00 ff ff ff ff ff ff ff ff ff ff ff 00 00 00 ff ff
ff ff ff ff 00 00 ff 00 00 ff 00 00 ff ff ff ff
ff ff ff 00 00 00 ff ff ff 00 00 ff ff 00 00 ff
00 00 ff 00 00 00 00 ff ff ff ff 00 00 00 00 00
ff ff ff ff ff 00 00 ff 00 00 ff ff ff ff ff ff
00 00 ff 00 00 00 00 00 ff ff ff ff ff 00 00 00
00 ff ff 00 00 ff ff ff 00 00 ff 00 00 00 ff ff
ff 00 00 ff 00 00 ff ff ff ff 00 00 ff ff 00 00
ff ff ff 00 00 00 ff ff ff ff ff ff ff ff ff ff
ff ff ff ff ff ff ff ff ff 00 00 00 00 00
```

- Type d'image ?
- Nb d'octets ?
- Offset ?
- Largeur ?
- Hauteur ?
- Nb octets/pixel ?
- Taille image ?
- Que représente-elle ?
- Taille (en ko) ?

# Exercice : ceci est une image ...

```

42 4d de 00 00 00 00 00 00 00 00 36 00 00 00 28 00
00 00 07 00 00 00 07 00 00 00 01 00 18 00 00 00
00 00 a8 00 00 00 13 0b 00 00 13 0b 00 00 00 00
00 00 00 00 00 00 ff ff ff ff ff ff ff ff ff 00
00 ff ff ff ff ff ff ff ff ff ff 00 00 00 ff ff
ff ff ff ff 00 00 ff 00 00 ff 00 00 ff ff ff ff
ff ff ff 00 00 00 ff ff ff 00 00 ff ff 00 00 ff
00 00 ff 00 00 00 00 ff ff ff ff 00 00 00 00 00
ff ff ff ff ff 00 00 ff 00 00 ff ff ff ff ff ff
ff 00 00 ff 00 00 ff ff ff ff 00 00 ff ff 00 00
ff ff ff 00 00 00 ff ff ff ff ff ff ff ff ff
ff ff ff ff ff ff ff ff ff 00 00 00 00 00
  
```

- BM car  $42_H = 66_D = B_{ASCII}$  et  $4d_H = 77_D = M_{ASCII}$
- Nb d'octets ?
- Offset ?
- Largeur ?
- Hauteur ?
- Nb octets/pixel ?
- Taille image ?
- Que représente-elle ?
- Taille (ko) ?

# Exercice : ceci est une image ...

```

42 4d de 00 00 00 00 00 00 00 00 36 00 00 00 28 00
00 00 07 00 00 00 07 00 00 00 01 00 18 00 00 00
00 00 a8 00 00 00 13 0b 00 00 13 0b 00 00 00 00
00 00 00 00 00 00 ff ff ff ff ff ff ff ff ff 00
00 ff ff ff ff ff ff ff ff ff ff 00 00 00 ff ff
ff ff ff ff 00 00 ff 00 00 ff 00 00 ff ff ff ff
ff ff ff 00 00 00 ff ff ff 00 00 ff ff 00 00 ff
00 00 ff 00 00 00 00 ff ff ff ff 00 00 00 00 00
ff ff ff ff ff 00 00 ff 00 00 ff ff ff ff ff ff
00 ff 00 00 ff 00 00 ff ff ff ff 00 00 ff ff 00 00
ff ff ff 00 00 00 ff ff ff ff ff ff ff ff ff ff
ff ff ff ff ff ff ff ff ff 00 00 00 00 00
  
```

- BM car  $42_H = 66_D = B_{ASCII}$  et  $4d_H = 77_D = M_{ASCII}$
- $de_H = 222_D$  octets
- Offset ?
- Largeur ?
- Hauteur ?
- Nb octets/pixel ?
- Taille image ?
- Que représente-elle ?
- Taille (ko) ?

# Exercice : ceci est une image ...

```

42 4d de 00 00 00 00 00 00 00 00 36 00 00 00 28 00
00 00 07 00 00 00 07 00 00 00 01 00 18 00 00 00
00 00 a8 00 00 00 13 0b 00 00 13 0b 00 00 00 00
00 00 00 00 00 00 ff ff ff ff ff ff ff ff ff 00
00 ff ff ff ff ff ff ff ff ff ff 00 00 00 ff ff
ff ff ff ff 00 00 ff 00 00 ff 00 00 ff ff ff ff
ff ff ff 00 00 00 ff ff ff 00 00 ff ff 00 00 ff
00 00 ff 00 00 00 00 ff ff ff ff 00 00 00 00 00
ff ff ff ff ff 00 00 ff 00 00 ff ff ff ff ff ff
ff 00 00 ff 00 00 ff ff ff ff 00 00 ff ff 00 00
ff ff ff 00 00 00 ff ff ff ff ff ff ff ff ff ff
ff ff ff ff ff ff ff ff ff 00 00 00 00 00
  
```

- BM car  $42_H = 66_D = B_{ASCII}$  et  $4d_H = 77_D = M_{ASCII}$
- $de_H = 222_D$  octets
- $36_H = 54_D$  octets
- Largeur ?
- Hauteur ?
- Nb octets/pixel ?
- Taille image ?
- Que représente-elle ?
- Taille (ko) ?



# Exercice : ceci est une image ...

```

42 4d de 00 00 00 00 00 00 00 36 00 00 00 28 00
00 00 07 00 00 00 07 00 00 00 01 00 18 00 00 00
00 00 a8 00 00 00 13 0b 00 00 13 0b 00 00 00 00
00 00 00 00 00 00 ff ff ff ff ff ff ff ff ff 00
00 ff ff ff ff ff ff ff ff ff ff 00 00 00 ff ff
ff ff ff ff 00 00 ff 00 00 ff 00 00 ff ff ff ff
ff ff ff 00 00 00 ff ff ff 00 00 ff ff 00 00 ff
00 00 ff 00 00 00 00 ff ff ff ff 00 00 00 00 00
ff ff ff ff ff 00 00 ff 00 00 ff ff ff ff ff ff
00 ff 00 00 ff 00 00 ff ff ff ff 00 00 ff ff 00 00
ff ff ff 00 00 00 ff ff ff ff ff ff ff ff ff ff
ff ff ff ff ff ff ff ff ff 00 00 00 00 00
  
```

- BM car  $42_H = 66_D = B_{ASCII}$  et  $4d_H = 77_D = M_{ASCII}$
- $de_H = 222_D$  octets
- $36_H = 54_D$  octets
- $07_H = 7$  pixels
- Hauteur ?
- Nb octets/pixel ?
- Taille image ?
- Que représente-elle ?
- Taille (ko) ?

# Exercice : ceci est une image ...

```

42 4d de 00 00 00 00 00 00 00 36 00 00 00 28 00
00 00 07 00 00 00 07 00 00 00 01 00 18 00 00 00
00 00 a8 00 00 00 13 0b 00 00 13 0b 00 00 00 00
00 00 00 00 00 00 ff ff ff ff ff ff ff ff ff 00
00 ff ff ff ff ff ff ff ff ff ff 00 00 00 ff ff
ff ff ff ff 00 00 ff 00 00 ff 00 00 ff ff ff ff
ff ff ff 00 00 00 ff ff ff 00 00 ff ff 00 00 ff
00 00 ff 00 00 00 00 ff ff ff ff 00 00 00 00 00
ff ff ff ff ff 00 00 ff 00 00 ff ff ff ff ff ff
00 00 ff 00 00 00 00 ff ff ff ff ff 00 00 00
ff 00 00 ff 00 00 ff ff ff ff 00 00 ff ff 00 00
ff ff ff 00 00 00 ff ff ff ff ff ff ff ff ff
ff ff ff ff ff ff ff ff ff 00 00 00 00 00
  
```

- BM car  $42_H = 66_D = B_{ASCII}$  et  $4d_H = 77_D = M_{ASCII}$
- $de_H = 222_D$  octets
- $36_H = 54_D$  octets
- $07_H = 7$  pixels
- $07_H = 7$  pixels
- Nb octets/pixel ?
- Taille image ?
- Que représente-elle ?
- Taille (ko) ?

# Exercice : ceci est une image ...

```

42 4d de 00 00 00 00 00 00 00 36 00 00 00 28 00
00 00 07 00 00 00 07 00 00 00 01 00 18 00 00 00
00 00 a8 00 00 00 13 0b 00 00 13 0b 00 00 00 00
00 00 00 00 00 00 ff ff ff ff ff ff ff ff ff 00
00 ff ff ff ff ff ff ff ff ff ff 00 00 00 ff ff
ff ff ff ff 00 00 ff 00 00 ff 00 00 ff ff ff ff
ff ff ff ff 00 00 00 ff ff ff 00 00 ff ff 00 00
ff ff ff ff 00 00 00 ff ff ff ff ff 00 00 00 00
ff ff ff ff 00 00 ff ff ff ff ff 00 00 00 ff ff
ff 00 00 ff 00 00 ff ff ff ff 00 00 ff ff 00 00
ff ff ff 00 00 00 ff ff ff ff ff ff ff ff ff ff
ff ff ff ff ff ff ff ff ff 00 00 00 00 00
  
```

- BM car  $42_H = 66_D = B_{ASCII}$  et  $4d_H = 77_D = M_{ASCII}$
- $de_H = 222_D$  octets
- $36_H = 54_D$  octets
- $07_H = 7$  pixels
- $07_H = 7$  pixels
- $18_H = 24_D$  bits/pixel, soit 3 octets/pixel
- Taille image ?
- Que représente-elle ?
- Taille (ko) ?

# Exercice : ceci est une image ...

```

42 4d de 00 00 00 00 00 00 00 00 36 00 00 00 28 00
00 00 07 00 00 00 07 00 00 00 01 00 18 00 00 00
00 00 a8 00 00 00 13 0b 00 00 13 0b 00 00 00 00
00 00 00 00 00 00 ff ff ff ff ff ff ff ff ff 00
00 ff ff ff ff ff ff ff ff ff ff 00 00 00 ff ff
ff ff ff ff 00 00 ff 00 00 ff 00 00 ff ff ff ff
ff ff ff ff 00 00 ff ff ff 00 00 ff ff 00 00 ff
00 00 ff 00 00 00 00 ff ff ff ff ff 00 00 00
00 ff ff 00 00 ff ff ff 00 00 ff 00 00 00 ff ff
ff 00 00 ff 00 00 ff ff ff ff 00 00 ff ff 00 00
ff ff ff 00 00 00 ff ff ff ff ff ff ff ff ff ff
ff ff ff ff ff ff ff ff ff 00 00 00 00 00
  
```

- BM car  $42_H = 66_D = B_{ASCII}$  et  $4d_H = 77_D = M_{ASCII}$
- $de_H = 222_D$  octets
- $36_H = 54_D$  octets
- $07_H = 7$  pixels
- $07_H = 7$  pixels
- $18_H = 24_D$  bits/pixel, soit 3 octets/pixel
- $a8_H = 168_D$  octets
- Que représente-elle ?
- Taille (ko) ?

# Exercice : ceci est une image ...

```

42 4d de 00 00 00 00 00 00 00 36 00 00 00 28 00
00 00 07 00 00 00 07 00 00 00 01 00 18 00 00 00
00 00 a8 00 00 00 13 0b 00 00 13 0b 00 00 00 00
00 00 00 00 00 00 ff ff ff ff ff ff ff ff ff 00
00 ff ff ff ff ff ff ff ff ff ff ff 00 00 00 ff ff
ff ff ff ff 00 00 ff 00 00 ff 00 00 ff ff ff ff
ff ff ff 00 00 00 ff ff ff 00 00 ff ff 00 00 ff
00 00 ff 00 00 00 00 ff ff ff ff 00 00 00 00 00
ff ff ff ff 00 00 ff 00 00 ff ff ff ff 00 00 ff ff
ff 00 00 ff 00 00 ff ff ff ff 00 00 ff ff 00 00
ff ff ff 00 00 00 ff ff ff ff ff ff ff ff ff ff
ff ff ff ff ff ff ff ff ff 00 00 00 00 00
  
```

- BM car  $42_H = 66_D = B_{ASCII}$  et  $4d_H = 77_D = M_{ASCII}$
- $de_H = 222_D$  octets
- $36_H = 54_D$  octets
- $07_H = 7$  pixels
- $07_H = 7$  pixels
- $18_H = 24_D$  bits/pixel, soit 3 octets/pixel
- $a8_H = 168_D$  octets
- Un cœur percé.
- Taille (ko) ?

# Exercice : ceci est une image ...

```

42 4d de 00 00 00 00 00 00 00 00 36 00 00 00 28 00
00 00 07 00 00 00 07 00 00 00 01 00 18 00 00 00
00 00 a8 00 00 00 13 0b 00 00 13 0b 00 00 00 00
00 00 00 00 00 00 ff ff ff ff ff ff ff ff ff 00
00 ff ff ff ff ff ff ff ff ff ff ff 00 00 00 ff ff
ff ff ff ff 00 00 ff 00 00 ff 00 00 ff ff ff ff
ff ff ff 00 00 00 ff ff ff 00 00 ff ff 00 00 ff
00 00 ff 00 00 00 00 ff ff ff ff 00 00 00 00
ff ff 00 00 ff ff ff 00 00 ff 00 00 00 ff ff
ff 00 00 ff 00 00 ff ff ff ff 00 00 ff ff 00 00
ff ff ff 00 00 ff ff ff ff ff ff ff ff ff ff
ff ff ff ff ff ff ff ff ff 00 00 00 00 00
  
```

- BM car  $42_H = 66_D = B_{ASCII}$  et  $4d_H = 77_D = M_{ASCII}$
- $de_H = 222_D$  octets
- $36_H = 54_D$  octets
- $07_H = 7$  pixels
- $07_H = 7$  pixels
- $18_H = 24_D$  bits/pixel, soit 3 octets/pixel
- $a8_H = 168_D$  octets
- Un cœur percé.
- 222 octets, soit 0,217 ko

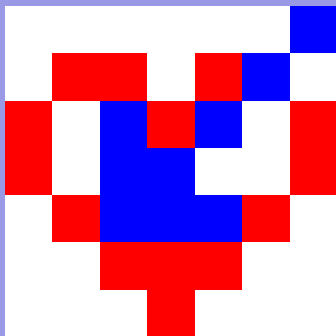
# Exercice : ceci est une image ...

```

42 4d de 00 00 00 00 00 00 00 36 00 00 00 28 00
00 00 07 00 00 00 07 00 00 00 01 00 18 00 00 00
00 00 a8 00 00 00 13 0b 00 00 13 0b 00 00 00 00
00 00 00 00 00 00 ff ff ff ff ff ff ff ff ff 00
00 ff ff ff ff ff ff ff ff ff ff 00 00 00 ff ff
ff ff ff ff 00 00 ff 00 00 ff 00 00 ff ff ff ff
ff ff ff 00 00 00 ff ff ff 00 00 ff ff 00 00 ff
00 00 ff 00 00 00 00 ff ff ff ff 00 00 00 00
ff ff 00 00 ff ff ff 00 00 ff 00 00 00 ff ff
ff 00 00 ff 00 00 ff ff ff ff 00 00 ff ff 00 00
ff ff ff 00 00 00 ff ff ff ff ff ff ff ff ff
ff ff ff ff ff ff ff ff ff 00 00 00 00 00
  
```

- BM car  $42_H = 66_D = B_{ASCII}$  et  $4d_H = 77_D = M_{ASCII}$
- $de_H = 222_D$  octets
- $36_H = 54_D$  octets
- $07_H = 7$  pixels
- $07_H = 7$  pixels
- $18_H = 24_D$  bits/pixel, soit 3 octets/pixel
- $a8_H = 168_D$  octets
- Un cœur percé.
- 222 octets, soit 0,217 ko

# Une autre interprétation de cette image



C'est plus naturel.



# Une autre image . . . corrompue.

```

42 4d e1 00 00 00 00 00 00 00 36 00 00 00 28 00
00 00 07 00 00 00 07 00 00 00 01 00 17 00 00 00
00 00 aa 00 00 00 13 0b 00 00 13 0b 00 00 00 00
00 00 00 00 00 00 00 ff 00 00 87 00 ff ff ff ff
ff ff ff ff ff ff ff ff ff ff ff ff 00 00 10 00 ff
00 00 87 00 ff ff ff ff ff ff ff ff ff ff ff ff
ff ff ff 00 00 00 00 ff 00 00 83 00 00 87 00 ff
ff ff ff ff ff ff ff ff ff ff ff 00 00 00 00 ff
00 00 ff 00 00 ff 00 00 ff 00 ff ff ff ff ff ff
ff ff ff 00 00 00 00 ff 00 00 86 00 ff ff ff ff
ff ff ff ff ff ff ff ff ff ff ff 00 00 00 00 ff
00 00 87 00 00 87 00 00 87 00 00 87 00 00 87 00
ff ff ff 00 00 00 00 ff 00 00 ff 00 00 ff 00 00
ff 00 00 ff 00 00 ff 00 00 ff 00 00 00 00

```

Ce fichier bichrome en couleurs vraies a été corrompu.

Que représente-t-il ?

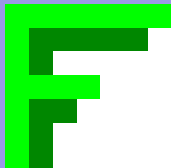
Pouvez-vous le corriger ?

Quel est le poids de cette image en Mo ?

# Le fichier une fois corrigé.

```

42 4d de 00 00 00 00 00 00 00 00 36 00 00 00 28 00
00 00 07 00 00 00 07 00 00 00 01 00 18 00 00 00
00 00 a8 00 00 00 13 0b 00 00 13 0b 00 00 00 00
00 00 00 00 00 00 00 ff 00 00 87 00 ff ff ff ff
ff ff ff ff ff ff ff ff ff ff ff 00 00 00 00 ff
00 00 87 00 ff ff ff ff ff ff ff ff ff ff ff ff
ff ff ff 00 00 00 00 ff 00 00 87 00 00 87 00 ff
ff ff ff ff ff ff ff ff ff ff ff ff 00 00 00 ff
00 00 ff 00 00 ff 00 00 ff 00 ff ff ff ff ff ff
ff ff ff 00 00 00 00 ff 00 00 87 00 ff ff ff ff
ff ff ff ff ff ff ff ff ff ff ff 00 00 00 00 ff
00 00 87 00 00 87 00 00 87 00 00 87 00 00 87 00
ff ff ff 00 00 00 00 ff 00 00 ff 00 00 ff 00 00
ff 00 00 ff 00 00 ff 00 00 ff 00 00 00 00
  
```



Nombre total d'octet (deux chiffres hexadécimaux) :

$$14 \cdot 16 - 2 = 222_D = de_H$$

Nombre de bits par pixel en vraie couleurs :

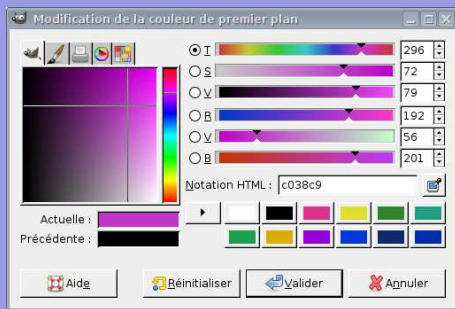
$$24_D = 18_H$$

Taille image 7 pixel sur 7 pixel (3 octets par pixel avec un nombre d'octets multiple de 4 par ligne  $\Rightarrow$  3 octets suppl. par ligne) :

$$7 \cdot 7 + 3 \cdot 7 = 168_D = a8_H$$

00 dans les octets supplémentaires par ligne.  
87 pour la couleur verte sombre (bichromie).

# Le modèle RVB



**R** 8 bits, 1 octet → 256  
niveaux de rouge

(0 pas de rouge, 255 rouge pur)

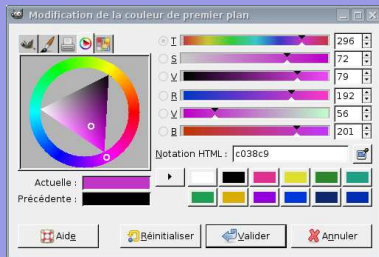
**V** 8 bits, 1 octet → 256  
niveaux de vert

(0 pas de vert, 255 vert pur)

**B** 8 bits, 1 octet → 256  
niveaux de bleu

(0 pas de bleu, 255 bleu pur)

# Le modèle Teinte-Saturation-Valeur

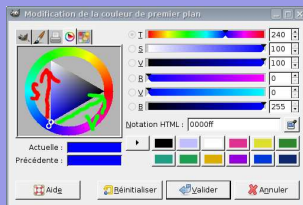
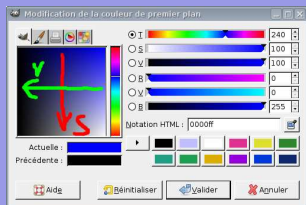


**Teinte** correspond à la couleur sur  $360^\circ$ .  
Le rouge se situe à  $0^\circ$ , le vert à  $120^\circ$  et le bleu à  $240^\circ$  selon un angle trigonométrique.

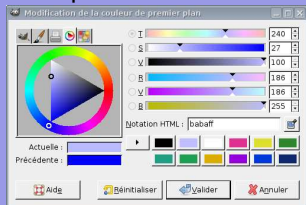
**Saturation** correspond à la quantité de couleur "dans" le blanc.

**Valeur** correspond, quant à elle, à la quantité de couleur "dans" le noir.

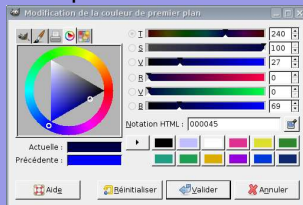
# Exemples



## Bleu peu saturé



## Bleu peu valorisé



# Mathématiquement

Si on considère  $M$  et  $m$ , respectivement, comme le maximum et le minimum des valeurs de l'ensemble  $(R, V, B)$ , les valeurs  $T$ ,  $S$  et  $V$  sont données par l'ensemble des relations ci-dessous où  $T \in [0, 360]$ .

$$T = \begin{cases} 0 & \text{si } M = m \\ 60 \cdot \frac{V-B}{M-m} + 0^\circ & \text{si } M = R \text{ et } V \geq B \\ 60 \cdot \frac{V-B}{M-m} + 360^\circ & \text{si } M = R \text{ et } V < B \\ 60 \cdot \frac{B-R}{M-m} + 120^\circ & \text{si } M = V \\ 60 \cdot \frac{R-V}{M-m} + 240^\circ & \text{si } M = B \end{cases}$$

$$S = \begin{cases} 0 & \text{si } M = 0 \\ (1 - \frac{m}{M}) \cdot 100 & \text{sinon} \end{cases}$$

$$V = \frac{M}{255} \cdot 100$$

# Exercices

Trouvez les valeurs TSV des couleurs en RVB suivantes et décrivez la couleur :

(34,56,239) ;

(22,240,200) ;

(255,231,176).

# Exercices

Trouvez les valeurs TSV des couleurs en RVB suivantes et décrivez la couleur :

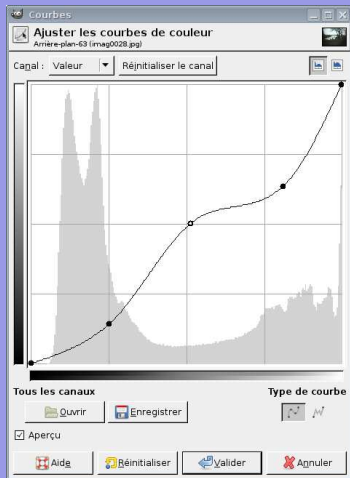
$(34,56,239) \rightarrow (234,86,94)$  **bleu très saturé et valorisé** ;

$(22,240,200) \rightarrow (169,91,94)$  **bleu-vert très saturé et valorisé** ;

$(255,231,176) \rightarrow (42,31,100)$  **orange peu saturé et très valorisé**.



# L'outils courbe



Description :

**Abscisse** : Niveaux de gris pour la valeur d'entrée.

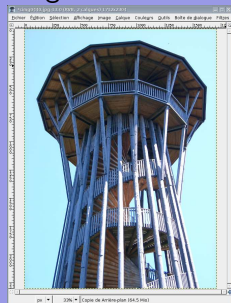
**Ordonnée** : Nombre de pixels de la valeur donnée.

La valeur de sortie correspond à la valeur d'entrée si la courbe est une droite diagonale.

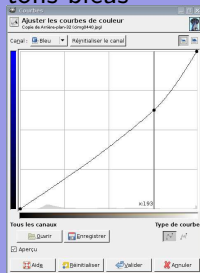
A gauche se trouve le *point noir* et à droite le *point blanc*.

Outil courbe : correction d'une dominante → pointer un pixel devant être gris, puis imposer RVB identiques

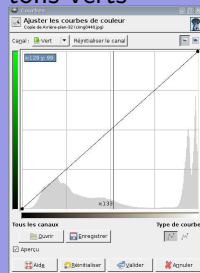
Original



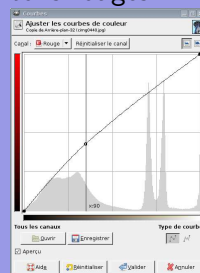
Répartition des tons bleus



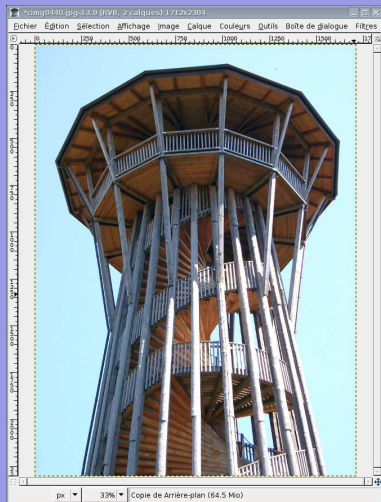
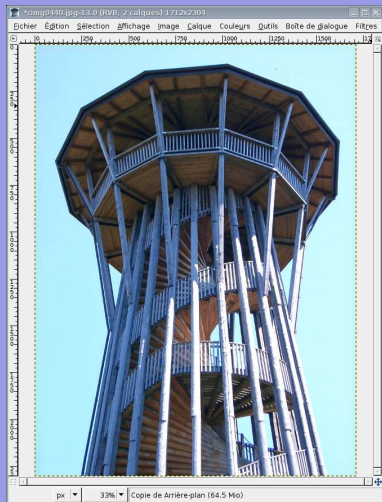
Répartition des tons verts



Répartition des tons rouges



# Original et correction de dominante



# Compression RLE : non destructive

## CODAGE PAR PLAGE OU RUN-LENGTH ENCODING.

Chaque octet (deux nombres hexadécimaux) est codé par le nombre de ses occurrences successives et sa valeur.

La chaîne ...

00 00 00 0A 12 00 00 23 23 23 23 12 12 AB AB AB AB AB

est alors codée de la manière suivante :

03 00 01 0A 01 12 02 00 04 23 02 12 05 AB

On a en effet, 03 octets 00, 01 octet 0A, 01 octet 12, 02 octets 00, 04 octets 23, 02 octets 12 et 05 octets AB.

# RLE : inconvénients

Plus il existe de plages identiques, plus la compression est importante.

Par contre, pour des images très détaillées, ce codage peut augmenter la taille de l'image :

## Un effet regrettable

01 00 AE AB 1C 3F → 01 01 01 00 01 AE 01 AB 01 1C 01 3F

Adapté à de grands aplats de couleurs identiques et aux images monochrome ou en un nombre restreint de couleurs (256 par exemple). Utilisé pour les icônes ou les documents au trait.

# Compression de Huffman : non destructive

**Idee : coder les lettres fréquentes par des bits de poids faibles.**

Une chaîne contenant cinq lettres

ssionnansnni ← oaisn

Pour coder cinq lettres, il faut au minimum trois bits :  $2^3 = 8$ .

Un codage peu intelligent

$o \rightarrow 000$  ;  $a \rightarrow 001$  ;  $i \rightarrow 010$  ;  $s \rightarrow 011$  ;  $n \rightarrow 100$

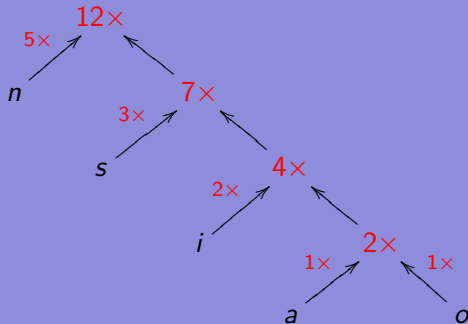
Le résultat est une chaîne de  $12 \cdot 3 = 36$  bits pour douze caractères.

La chaîne encodée

011|011|010|000|100|100|001|100|011|100|100|010

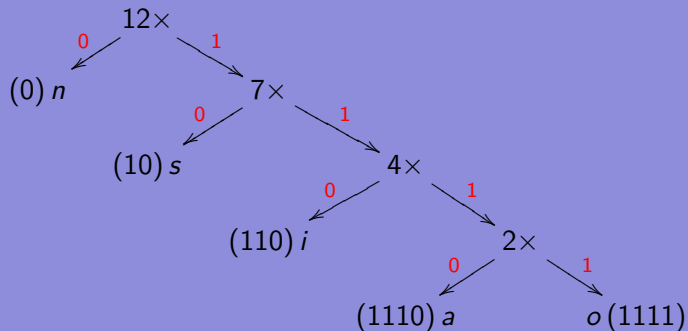
# Méthode : éléments de plus basse fréquence 2 à 2.

## Arbre de Huffman : construction



Méthode : 1 sur bit fort pour éléments basse fréquence.

### Arbre de Huffman : codage





# Codage de Huffman

Selon l'arbre de Huffman, la lettre n est codée 0, la lettre s est codée 10, la lettre i est codée 110, la lettre a est codée 1110 et la lettre o codée 1111

## Codage de Huffman

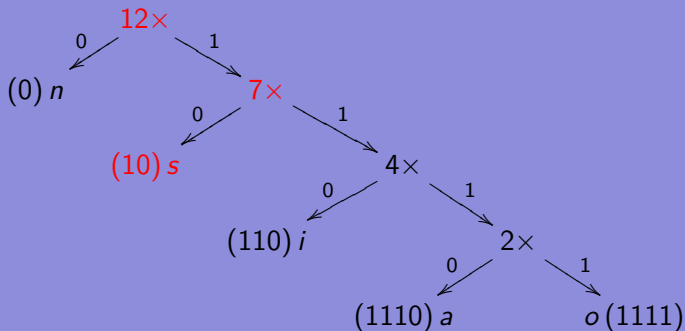
ssionnansnni  $\rightarrow$  10|10|110|1111|0|0|1110|0|10|0|0|110

soit 25 bits au lieu de 36 ! Une compression de 30% !

De plus, le codage permet la suppression des bits nuls de poids fort. Le codage est UNIVOQUE.

Méthode : on lit simplement en suivant l'arbre.

Arbre de Huffman : **décodage** 1010110...



# Exercices

Déterminez le poids de l'image du double F précédemment présentée, mais codée selon RLE puis selon Huffman.

# Compression RLE pour le double F

```

42 4d de 00 00 00 00 00 00 00 00 36 00 00 00 28 00
00 00 07 00 00 00 07 00 00 00 01 00 18 00 00 00
00 00 a8 00 00 00 13 0b 00 00 13 0b 00 00 00 00
00 00 00 00 00 00 00 ff 00 00 87 00 ff ff ff ff
ff ff ff ff ff ff ff ff ff ff ff ff 00 00 00 00 ff
00 00 87 00 ff ff ff ff ff ff ff ff ff ff ff ff ff
ff ff ff 00 00 00 00 ff 00 00 87 00 00 87 00 ff
ff ff ff ff ff ff ff ff ff ff ff ff 00 00 00 00 ff
00 00 ff 00 00 ff 00 00 ff 00 ff ff ff ff ff ff ff
ff ff ff 00 00 00 00 ff 00 00 87 00 ff ff ff ff ff
ff ff ff ff ff ff ff ff ff ff ff ff 00 00 00 00 ff
00 00 87 00 00 87 00 00 87 00 00 87 00 00 87 00
ff ff ff 00 00 00 00 ff 00 00 ff 00 00 ff 00 00
ff 00 00 ff 00 00 ff 00 00 ff 00 00 00 00
  
```

```

01 42 01 4d 01 de 07 00 01 36 03 00 01 28 03 00
01 07 03 00 01 07 03 00 01 01 01 00 01 18 05 00
01 a8 03 00 01 13 01 0b 02 00 01 13 01 0b 0b 00
01 FF 02 00 01 87 01 00 0F FF 04 00 01 FF 02 00
01 87 01 00 0F FF 04 00 01 FF 02 00 01 87 02 00
01 87 01 00 0C FF 04 00 01 FF 02 00 01 FF 02 00
01 FF 02 00 01 FF 01 00 09 FF 04 00 01 FF 02 00
01 87 01 00 0F FF 04 00 01 FF 02 00 01 87 02 00
01 87 02 00 01 87 02 00 01 87 02 00 01 87 01 00
03 FF 04 00 01 FF 02 00 01 FF 02 00 01 FF 02 00
01 FF 02 00 01 FF 02 00 01 FF 02 00 01 FF 04 00
  
```

$16 \times 11 = 176$  octets

$16 \times 14 - 2 = 222$  octets

$$\text{compression : } c = \frac{222 - 176}{222} \cdot 100 = 20,7\%$$

# Compression de Huffman pour le double F

```

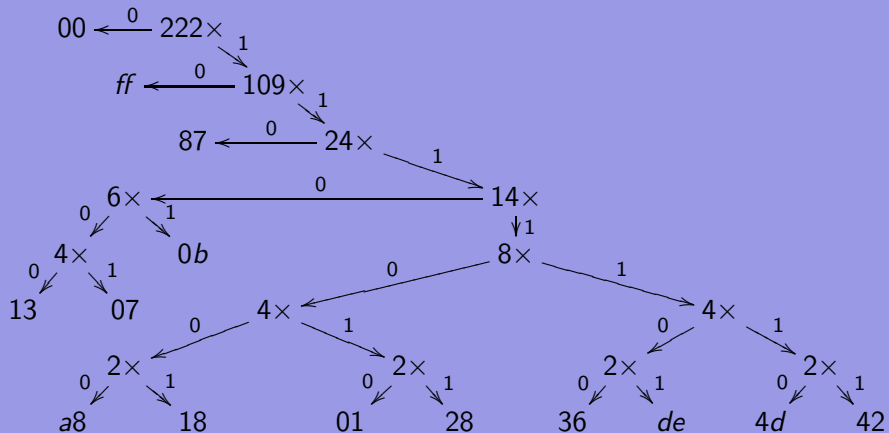
42 4d de 00 00 00 00 00 00 00 00 36 00 00 00 28 00
00 00 07 00 00 00 07 00 00 00 01 00 18 00 00 00
00 00 a8 00 00 00 13 0b 00 00 13 0b 00 00 00 00
00 00 00 00 00 00 00 ff 00 00 87 00 ff ff ff ff
ff ff ff ff ff ff ff ff ff ff ff 00 00 00 00 ff
00 00 87 00 ff ff ff ff ff ff ff ff ff ff ff ff
ff ff ff 00 00 00 00 ff 00 00 87 00 00 87 00 ff
ff ff ff ff ff ff ff ff ff ff ff 00 00 00 00 ff
00 00 ff 00 00 ff 00 00 ff 00 ff ff ff ff ff ff
ff ff ff 00 00 00 00 ff 00 00 87 00 ff ff ff ff
ff ff ff ff ff ff ff ff ff ff ff 00 00 00 00 ff
00 00 87 00 00 87 00 00 87 00 00 87 00 00 87 00
ff ff ff 00 00 00 00 ff 00 00 ff 00 00 ff 00 00
ff 00 00 ff 00 00 ff 00 00 ff 00 00 00
  
```

$16 \times 14 - 2 = 222$  octets

Octets	Fréquence
42, 4d, de, 36,	1
28, 01, 18, a8	
07, 13, 0b	2
87	10
ff	85
00	113

14 symboles :  $2^4 = 16 \Rightarrow 4$  bits

# Arbre de Huffman



# Codage de Huffman

Arbre binaire  $\Rightarrow$  14 symboles sur 7 bits !

Octets	Fréquence	codage	Total
42, 4d, de, 36, 28, 01, 18, a8	1	7 bits $\Rightarrow 8 \cdot 7 =$	56 bits
07, 13, 0b	2	6 bits $\Rightarrow 3 \cdot 2 \cdot 6 =$	36 bits
87	10	3 bits $\Rightarrow 10 \cdot 3 =$	30 bits
ff	85	2 bits $\Rightarrow 85 \cdot 2 =$	170 bits
00	113	1 bits $\Rightarrow 113 \cdot 1 =$	113 bits
		total : 405 bits =	51 octets

$$\text{Compression : } c = \frac{222 - 51}{222} \cdot 100 = 77\%$$

# Principaux formats d'image : TIFF

- **TIFF ou Tag Image File Format** : professionnels de l'image.
- Non compressé : images sans perte de qualité.
- Compressé (LZW (Lempel-Ziv-Welch) pour images ayant peu de couleurs ou d'autres ...) sans perte de définition (non destructif).
- Taille importante : pas recommandé pour une utilisation sur le web.
- Format propriétaire (Adobe, Adobe). Avec le JPG et malgré qu'il soit propriétaire, c'est le format de numérisation du ministère de la culture en France.



# Principaux formats d'image : JPG

- **JPG ou Joint Photographic Expert Group** : format du web.
- Compressé en fonction de la qualité de l'image finale désirée (RLE, Huffmann et transformée de fourier).
- Pas un format propriétaire mais plutôt une norme publique, même si certains en revendiquent la paternité.
- Utilisé communément par les appareils photographique numérique.
- C'est LE format le plus utilisé aujourd'hui.

# Principaux formats d'image : GIF

- **GIF ou Graphics Interchange Format** : format 8 bits, c'est-à-dire avec au maximum 256 couleurs.
- Images sont donc de petite taille adaptées au web.
- Images de type logo ou dessin animés. Banderoles et autres petits dessins animés égayant les pages web.
- Ce format totalement propriétaire (CompuServe) est maintenant dans le domaine public depuis peu.

# Principaux formats d'image : PNG

- **PNG ou Portable Network Graphics** : destiné à remplacer progressivement le format GIF sur internet.
- Meilleur taux de compression (compression png) que le GIF.
- Format totalement libre.

# Principaux formats d'image : EPS

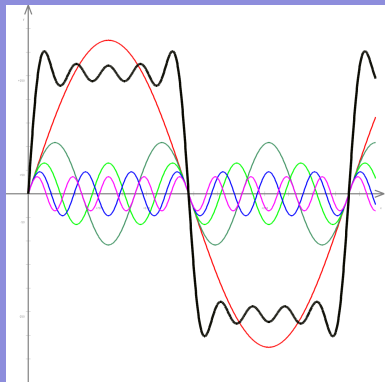
- **EPS ou Encapsuled PostScript** directement à une imprimante postscript.
- Gère des images de type bitmap (ensemble de points) ou vectorielle (des courbes et lignes).
- Comprend le RVB ou le CMJN.
- Pas un format propriétaire.
- Format d'image. Ne pas confondre avec le PostScript qui n'est pas à proprement parlé un format d'image, mais un langage d'instruction d'impression.

# Principaux formats d'image : PDF

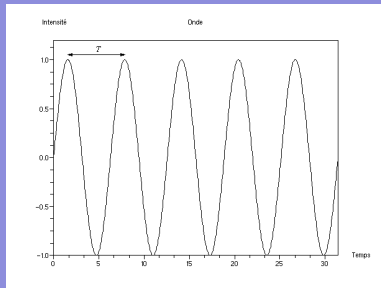
- **PDF ou Portable Document Format** visualisation et l'impression d'un document indépendamment de la plate-forme.
- Peu gourmand en espace et est donc intéressant pour le web.
- Non compressé ou compressé non destructif.
- Format propriétaire (Adobe), mais ouvert et parfaitement documenté.

# La physique du son

## Décomposition de Fourier<sup>27</sup>

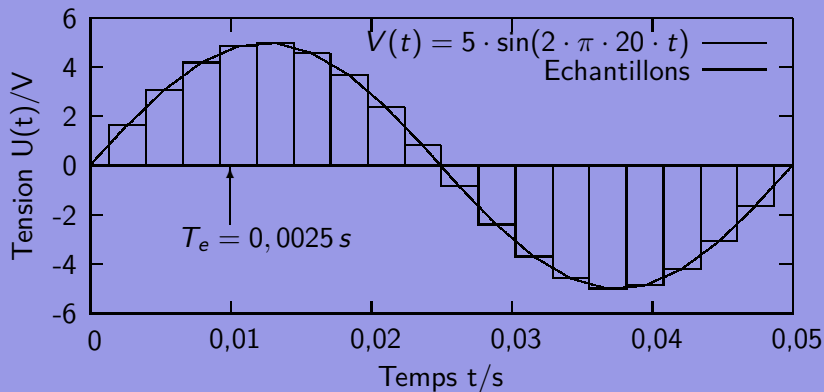


## Onde sinusoidale<sup>28</sup>



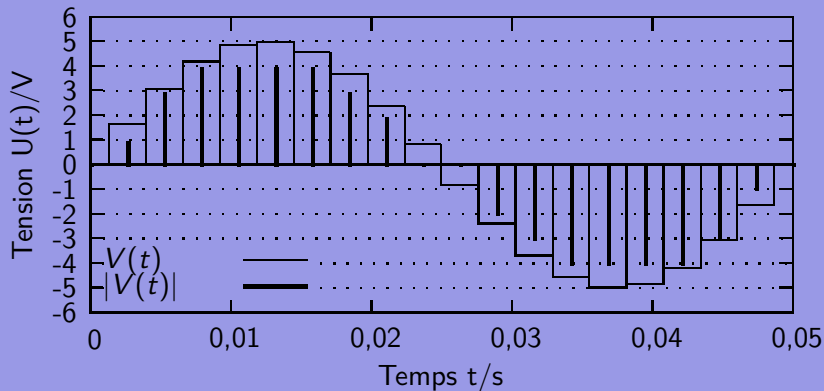
# Échantillonnage

Signal 20 Hz ; freq. echant. 400 Hz



# Quantification

Signal 20 Hz ; freq. echant. 400 Hz





# Numérisation du signal : histogramme et codage sur 3 bits

Attention, un codage sur trois bits introduit un fort bruit. Il faut, pour avoir un rendu correct de la musique, au minimum 16 bits de dynamique (quantification). Le codage hexadécimal est alors adéquat.

V(k)	1	2	3	4	5	6	7	8	9	10	11	12
déc	0	1	3	4	4	4	4	3	2	0	0	-2
bin	0	1	11	100	100	100	100	11	10	0	0	110
hex	0	1	3	4	4	4	4	3	2	0	0	?

# Grandeurs importantes

Plusieurs grandeurs sont à considérer :

- durée du morceau,
- fréquence maximale du signal
- fréquence d'échantillonnage (thm de Shannon) : double de la fréquence maximale suffit
- dynamique, nombre de bit de quantification
- nombre de canaux (2.0 : stéréo, pas de caisson de basse ; 5.1 : trois haut-parleurs avant, deux arrières et un caisson de basse)

# Formats audio-numériques

La gamme des fréquences audibles se situe entre 20 Hz et 20 kHz. Shannon implique donc :  $f_e \geq 40'000 \text{ Hz}$  (44'100 car une ligne PAL 294 points de trois couleurs à 50 images par seconde, soit  $294 \cdot 3 \cdot 50 = 44'100$ ).

Le flux ou "bitrate" se calcule par la multiplication de l'échantillonnage par la quantification (en kilobits/s ou kbps) et le nombre de canaux.

Support	Phone	DV	CD aud	DVD vid	DVD HD
Échant. (Hz)	8'000	48'000	44'100	48'000	96'000
Quantif. (bits)	8	16	16	16	24
Canaux	1	2	2	6	8
Flux (kbits/s)	62,5	1500	1378,1	4500	18000

# Fichiers, conteneurs, formats, extension, codecs et encodeur

## Fichier – conteneur



# Principaux formats et codecs audio

## WAV

Conteneur : mp3, pcm, wma . . . , mais par référence non compressé :  $10 \text{ Mo/min} = 44'100 \cdot 16 \cdot 2 \cdot 60/8/1024/1024$

## MP3

Compression destructive (ds hautes fréquences)  
Rapport 1/10 wav ; débit fixe  
1Mo/min (CD)  
Tag ID3

## OGG/VORBIS

Compression destructive ; mieux mp3  
Rapport 1/10 wav ; débit variable  
Tag ID3 ; libre ; encodeur oggenc

# Suite

## FLAC

Free Lossless Audio  
Compression sans pertes  
Rapport 1/7 wav  
Libre

## AAC

Advanced Audio Coding  
Compression destructive  
Rapport 1/10 wav ; Ipod, Itune  
Meilleur car MPEG-4 ; encodeur  
faac

## WMA

Windows Media Audio  
Compression destructive  
Rapport variable  
Dispositifs anticopie

# Table des matières

## 6 Annexes

- Motorisation télescope
- LibreOffice
- Évaluations
- Évaluations
- L<sup>A</sup>T<sub>E</sub>X

1 DF Systèmes numériques

2 DF Informations et données

3 DF Algo. et programmation

4 DF Informatique et société

## Exemple : “Un projet télescopique”

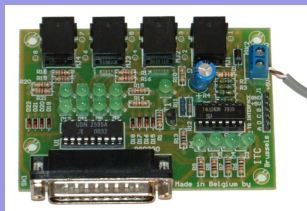
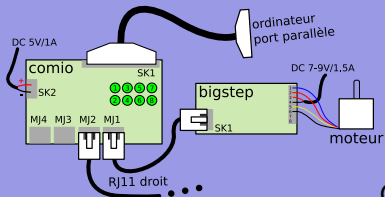
**Objectif** Réaliser le suivi d'une étoile guide par la motorisation du télescope du lycée.

**Moyens matériels** Deux moteurs, une carte de commande sur port parallèle, un ordinateur portable.

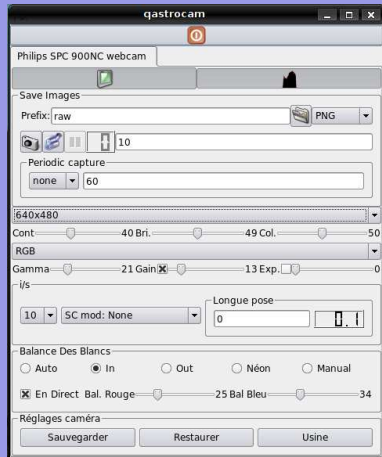
**Moyens logiciels** Aucun driver, un logiciel libre de suivi et de photographie astronomique pour webcam.

**Rejetés** Iris (fantastique logiciel gratuit de traitement d'images astro) car seuls les montures motorisées de type LX200 sont prévues. Code source fermé  
⇒ accord et implication du développeur.





C++



# Intégration du code

## Code source ajouté à gastrocam :

Qastrocam <http://3demi.net/astro/gastrocam/doc/>



```

1 #ifndef _QTelescopeLBC_hpp_
2 #define _QTelescopeLBC_hpp_
3 #include "QTelescope.hpp"
4 class PPort;
5 void *runEW(void *arg);
6 void *runSN(void *arg);
7 class QTelescopeLBC : public QTelescope {
8     Q_OBJECT;
9 public:
10     QTelescopeLBC(PPort * pport);
11     void goE(float shift);
12     void goW(float shift);
13     void goS(float shift);
14     void goN(float shift);
15     void stopE();
16     void stopW();
17     void stopS();
18     void stopN();
19     void threads_start();
20     void threads_stop();
21     double setSpeed(double speed);
22     bool setTracking(bool activated);
23 };
24 #endif

```

# Limites du projet

## Critiques importantes :

- Ouverture du code : bien  $\Rightarrow$  adaptation spécifique au télescope du lycée.  
Mais cette adaptation est versatile à cause des mises-à-jour  $\Rightarrow$  nécessité d'une documentation précise permettant la recompilation du logiciel modifié (impossible sans code source!).
- Pérérité de l'adaptation  $\Rightarrow$  soutient des développeurs (déchargés de produire l'adaptation).
- Reste toujours la possibilité de fork (projet indépendant).

[retour licences libres](#)

# Le contexte de LibreOffice

## Microprojet

Récupérer la page “informatique”  
de Wikipedia pour impression.

# Le contexte de LibreOffice

## Microprojet

Récupérer la page “informatique” de Wikipedia pour impression.

## Problèmes rencontrés

- Copier-coller : plantage OO-2.4.
- Copier-coller-texte : pas de structure.
- Table des matières incorrecte.
- Liens soulignés et en bleu.
- Notes de fin non actives.
- Références non actives.
- Image non référencée.

# Pas de problèmes, que des solutions

## Problèmes rencontrés

- Copier-coller : plantage OO-2.4.
- Copier-coller-texte : pas de structure.
- Table des matières incorrecte.
- Liens soulignés et en bleu.
- Notes de fin non actives.
- Références non actives.
- Image non référencée.

## Solutions

- OO-3 ; découper le document.

# Pas de problèmes, que des solutions

## Problèmes rencontrés

- Copier-coller : plantage OO-2.4.
- Copier-coller-texte : pas de structure.
- Table des matières incorrecte.
- Liens soulignés et en bleu.
- Notes de fin non actives.
- Références non actives.
- Image non référencée.

## Solutions

- OO-3 ; découper le document.
- Utiliser les styles.

# Pas de problèmes, que des solutions

## Problèmes rencontrés

- Copier-coller : plantage OO-2.4.
- Copier-coller-texte : pas de structure.
- **Table des matières incorrecte.**
- Liens soulignés et en bleu.
- Notes de fin non actives.
- Références non actives.
- Image non référencée.

## Solutions

- OO-3 ; découper le document.
- Utiliser les styles.
- Styles et index.



# Pas de problèmes, que des solutions

## Problèmes rencontrés

- Copier-coller : plantage OO-2.4.
- Copier-coller-texte : pas de structure.
- Table des matières incorrecte.
- Liens soulignés et en bleu.
- Notes de fin non actives.
- Références non actives.
- Image non référencée.

## Solutions

- OO-3 ; découper le document.
- Utiliser les styles.
- Styles et index.
- Styles de liens.

# Pas de problèmes, que des solutions

## Problèmes rencontrés

- Copier-coller : plantage OO-2.4.
- Copier-coller-texte : pas de structure.
- Table des matières incorrecte.
- Liens soulignés et en bleu.
- **Notes de fin non actives.**
- **Références non actives.**
- **Image non référencée.**

## Solutions

- OO-3 ; découper le document.
- Utiliser les styles.
- Styles et index.
- Styles de liens.
- Pénible ou XML.

# Pas de problèmes, que des solutions

## Problèmes rencontrés

- Copier-coller : plantage OO-2.4.
- Copier-coller-texte : pas de structure.
- Table des matières incorrecte.
- Liens soulignés et en bleu.
- Notes de fin non actives.
- Références non actives.
- Image non référencée.

## Solutions

- OO-3 ; découper le document.
- Utiliser les styles.
- Styles et index.
- Styles de liens.
- Pénible ou XML.
- Base de donnée OO

# Pas de problèmes, que des solutions

## Problèmes rencontrés

- Copier-coller : plantage OO-2.4.
- Copier-coller-texte : pas de structure.
- Table des matières incorrecte.
- Liens soulignés et en bleu.
- Notes de fin non actives.
- Références non actives.
- **Image non référencée.**

## Solutions

- OO-3 ; découper le document.
- Utiliser les styles.
- Styles et index.
- Styles de liens.
- Pénible ou XML.
- Base de donnée OO
- Légende : table des illustrations.

# LibreOffice, Xhtml, XML

Background de la mise en forme sous OO.

OO et XML

OO – XML

.odt  $\leftrightarrow$  .zip

# LibreOffice, Xhtml, XML

Background de la mise en forme sous OO.

OO et XML

OO – XML

.odt  $\leftrightarrow$  .zip

## Structure archive zip

`meta.xml` donne des informations générales sur le document : auteur, dates ...

`styles.xml` contient les styles utilisés,

`content.xml` contient le contenu du document,

`settings.xml` qui contient des informations relatives aux applications utilisées (imprimantes, ...).

# Exemple

Une structure classique :  
titres hiérarchisés.

<b>Les transports</b>
<i><b>Pour les jeunes</b></i>
<b>Le skate</b>
Planche à roulette
<b>Le roller</b>
Patins à roulette
<b>le vélo</b>
Deux roues
<i><b>Pour les vieux</b></i>
<b>La voiture</b>
Quatre roues
<b>Le bateau</b>
Une coque

# Traduction en XML

```
<office:body>
  <office:text>
    </text:sequence-decls>
    <text:h text:style-name="Heading_20_1" text:outline-level="1">Les transports<
      /text:h>
    <text:p text:style-name="Standard" />
    <text:h text:style-name="Heading_20_2" text:outline-level="2">Pour les jeunes
      </text:h>
    <text:h text:style-name="Heading_20_3" text:outline-level="3">Le skate</
      text:h>
    <text:p text:style-name="Standard">Planche a roulette</text:p>
    <text:h text:style-name="Heading_20_3" text:outline-level="3">Le roller</
      text:h>
    <text:p text:style-name="Standard">Patins a roulette</text:p>
    <text:h text:style-name="Heading_20_3" text:outline-level="3">le velo</text:h
      >
    <text:p text:style-name="Standard">Deux roues</text:p>
    <text:h text:style-name="Heading_20_2" text:outline-level="2">Pour les vieux<
      /text:h>
    <text:h text:style-name="Heading_20_3" text:outline-level="3">La voiture</
      text:h>
    <text:p text:style-name="Text_20_body">Quatre roues</text:p>
    <text:h text:style-name="Heading_20_3" text:outline-level="3">Le bateau</
      text:h>
    <text:p text:style-name="Text_20_body">Une coque</text:p>
  </office:text>
</office:body>
```



# Trois évaluations pour quatre notes

## TE CONNAISSANCES GÉNÉRALES

- Début décembre, impérativement avant Noël.
- La matière vue jusque là.

## Projet PROGRAMMATION (compte double)

- Validation,
- Rendre à date spécifiée,
- À deux (même note),
- Application particulière imposée,
- Documentation conséquente,
- Licence, respect du RGPD et du droit d'auteur (images et code),

## Projet informatique et société

- Par groupe de deux.
- Une page recto-verso de texte.

Retour cours

# Trois évaluations pour quatre notes

## Projet TEXTE

- Validation par l'enseignant
- Rendre avant Noël,
- À deux (même note),
- Un site en HTML,
- Une page expliquant les problèmes rencontrés, tout compte,
- Une licence expliquée.

## Projet JAVASCRIPT (compte double)

- Validation,
- Rendre à date spécifiée,
- À deux (même note),
- Site ou application javascript,
- Documentation conséquente,
- Licence, respect du RGPD et du droit d'auteur (images et code),

## Projet libre Une page recto-verso de texte.

[Retour cours](#)

# Structure générale

Fichier  
d'extension

.**tex**

Structure  
identique au  
fichier source  
d'un  
programme  
d'informatique.

## Préambule

```
\documentclass[a4paper,10pt]{book}  
\usepackage[options]{nom du paquet}
```

## Corps

```
\begin{document}  
...  
\end{document}
```

Attention, L<sup>A</sup>T<sub>E</sub>X n'est pas un langage XML !

# Le préambule

Voici un exemple de préambule commenté  
(en L<sup>A</sup>T<sub>E</sub>X les commentaires commencent par %).

```

\documentclass[a4paper,10pt]{book}
\usepackage[T1]{fontenc}           % Pour les accents
\usepackage[utf8]{inputenc}       % Pour la gestion de l'encodage
\usepackage[french]{babel}        % Pour la gestion de la typo française
\usepackage{verbatim}             % Pour mettre des commentaires étendus
\usepackage{graphicx}             % Pour mettre des images
\graphicspath{{./Images/}}       % Pour éviter de mettre le chemin des images
\usepackage{fancyhdr}             % Pour des entêtes
\fancyhead[RE,LO]{Lycee Blaise Cendrars\\La Chau-de-Fonds}
\fancyhead[LE,RO]{\thepage}
\lfoot{} \cfoot{} \rfoot{}
\pagestyle{fancy}                 % Pour un style de page particulier

\title{Galilee (savant)}          % A mettre dans le titre
\author{Un article de Wikipedia, l'encyclopedie libre.}

\setlength{\headheight}{22.6pt} % pour élargir un peu l'entête

```

# Le corps

Voici un exemple de structure du corps du document.

```
\begin{document}           % Debut du corps du document
\maketitle                 % Cree une page de titre
\begin{comment}           % Pour un commentaire long
...
\end{comment}
...
\tableofcontents{}        % Cree une table des matieres

\chapter{Premier chapitre} % Premier chapitre
\section{Sous-titre}      % Premiere section
...
\section{Sous-titre}      % Seconde section
...
\end{document}           % Fin du corps du document
```

# Quelques commandes de base

Évidemment, la richesse de L<sup>A</sup>T<sub>E</sub>X tient dans celle de ses commandes.

En voici quelques-unes :

Maths hors ligne :

```
\[E=m\cdot c^2\]
```

$$E = m \cdot c^2$$

Maths en ligne :

```
\(E=m\cdot c^2\)
```

$$E = m \cdot c^2$$

Liste :

```
\begin{itemize}
  \item Truc
  \item Chose
\end{itemize}
```

Le résultat :

- Truc
- Chose

# Table des matières

## 7 Internet

- Constituants des pages
- Les CMS : Systèmes de Management de Contenu
- HTML
- Animation javascript

1 DF Systèmes numériques

2 DF Informations et données

3 DF Algo. et programmation

4 DF Informatique et société

# Constituants des pages

Un site web est constitué de :



# Constituants des pages

Un site web est constitué de :

- Structure et contenu HTML ou XHTML

# Constituants des pages

Un site web est constitué de :

- Structure et contenu HTML ou XHTML
- Présentation avec les feuilles de style CSS

# Constituants des pages

Un site web est constitué de :

- Structure et contenu HTML ou XHTML
- Présentation avec les feuilles de style CSS
- Graphisme et intégration des images GIF, JPG, PNG, MNG.

# Constituants des pages

Un site web est constitué de :

- Structure et contenu HTML ou XHTML
- Présentation avec les feuilles de style CSS
- Graphisme et intégration des images GIF, JPG, PNG, MNG.
- Anciennement, animation en Flash, aujourd'hui avec attributs CSS3 ou SVG et javascript.

# Constituants des pages

Un site web est constitué de :

- Structure et contenu HTML ou XHTML
- Présentation avec les feuilles de style CSS
- Graphisme et intégration des images GIF, JPG, PNG, MNG.
- Anciennement, animation en Flash, aujourd'hui avec attributs CSS3 ou SVG et javascript.
- Incorporation de multimédias (sons, vidéos...).

# Constituants des pages

Un site web est constitué de :

- Structure et contenu HTML ou XHTML
- Présentation avec les feuilles de style CSS
- Graphisme et intégration des images GIF, JPG, PNG, MNG.
- Anciennement, animation en Flash, aujourd'hui avec attributs CSS3 ou SVG et javascript.
- Incorporation de multimédias (sons, vidéos...).
- Dynamisme au niveau de la gestion de contenu (coté serveur) avec des langages de développement de type PHP, Java ... fonctionnant avec un serveur Web.

# Constituants des pages

Un site web est constitué de :

- Structure et contenu HTML ou XHTML
- Présentation avec les feuilles de style CSS
- Graphisme et intégration des images GIF, JPG, PNG, MNG.
- Anciennement, animation en Flash, aujourd'hui avec attributs CSS3 ou SVG et javascript.
- Incorporation de multimédias (sons, vidéos...).
- Dynamisme au niveau de la gestion de contenu (coté serveur) avec des langages de développement de type PHP, Java ... fonctionnant avec un serveur Web.
- Des bases de données de type SQL ou XML.

# Côté client

La notion de client-serveur désigne un mode de communication dans un réseau, comme par exemple un réseau internet.



# Côté client

La notion de client-serveur désigne un mode de communication dans un réseau, comme par exemple un réseau internet.

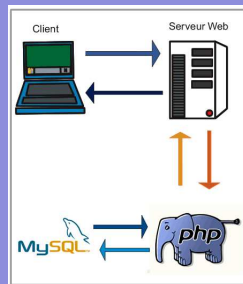
## Le client

Le client envoie des requêtes. Exemple, l'utilisateur internaute clique sur un lien. Il envoie donc une requête au serveur sur lequel le site est installé. Il passe par un navigateur qui envoie la requête, puis interprète les pages HTML que lui retourne le serveur pour les afficher.

# Côté serveur

## Le serveur

Le serveur attend les requêtes de l'utilisateur (serveur web), réunit ce qu'il est nécessaire pour composer une réponse (base de donnée), la construit (php) et la retourne au client.



# Beaucoup de choix

## Ne pas refaire le monde. Dynamique en php et javascript déjà présente.

Différents niveaux de pré-fabriqué, en fonction des besoins :

- 1 CMS déjà installé sur site. Un panneau de configuration → quelques interactions aspects visuels - contenu. Problème : publicités.
- 2 CMS à installer. Le contrôle du code → adaptation aux spécificités. Sécurité et mise-à-jour prises en charge par les développeurs du CMS.
- 3 CMS sans structure php. HTML, CSS et javascript → compatibilité navigateurs - mobiles. Problèmes : sécurité - mises-à-jour. Exemples : Initializr ou boilerplate (template).

# Un cms parmi d'autres : GetSimple

- libre et gratuit, sous licence GPL3,
- un code propre en HTML5 et CSS3,
- une gestion privée des utilisateurs par interface graphique,
- une gestion privée des documents,
- différents modèles de sites pré-faits,
- la création de pages, via un éditeur simple,
- des mises-à-jour régulières, ...

Tout cela nécessitant un serveur web, le langage php, mais pas de base de donnée SQL. Celle-ci est remplacée par un traitement XML des données.

# Installation

Installation simple par dépôt ftp (sous filezilla par exemple).

## FTP : File Transfert Protocol

Il s'agit d'un ensemble de règles de transfert d'information sur le réseau par copie de fichiers. On peut parler des outils nécessaires pour gérer des opérations de navigateur de fichiers sur des machines distantes.

# Résultat

## Une page d'accueil

The screenshot shows a CMS front-end page with a dark blue header. The main content area is white and contains a 'Bienvenue' section with placeholder text. A right-hand sidebar is titled 'LETSMILE FEATURES' and lists several items. The footer includes a copyright notice for '© 2009-2010 GetSmile CMS - Version 3.2.0'.

**TEST1** ACCUEIL SECONDE PAGE

### Bienvenue

Bla bla bla a load of old nonsense goes here in this area, move on nothing of interests to read here just a load of old nonsense Bla bla bla a load of old nonsense goes here in this area, move on nothing of interests to read here just a load of old nonsense.

**[W]**

Bla bla bla a load of old nonsense goes here in this area, move on nothing of interests to read here just a load of old nonsense Bla bla bla a load of old nonsense goes here in this area, move on nothing of interests to read here just a load of old nonsense.

Publication October 20th, 2010

**LETSMILE FEATURES**

- Full content area usage
- Drag and Drop interface
- User creation & backend
- Easy to manage
- Great documentation
- Growing community

This is your sidebar area. Please change me in Theme > Site Components

[Download the Latest GetSmile](#)

© 2009-2010 GetSmile CMS - Version 3.2.0

## Une zone d'administration

The screenshot shows a CMS back-end administration page with a dark blue header. The main content area is white and contains a 'Gestion des pages' section with a table of pages. A right-hand sidebar contains several action buttons. The footer includes a copyright notice for '© 2009-2010 GetSmile CMS - Version 3.2.0'.

**TEST1** Bienvenue général Administration

Pages Filaires Titres Catégories Plugins Support Configuration

### Gestion des pages

ATTENDEZ LE CHARGEMENT... [X] [Y]

TITRE DE LA PAGE	DATE
Bienvenue	20100913 11
- Catégorie_Bla	20100913 11

**Z page(s) en tout**

Gestion des articles • Gestion des fichiers • Gestion des thèmes • Gestion des commentaires • Gestion des pages • Paramètres système • Support • Paramètres

© 2009-2010 GetSmile CMS - Version 3.2.0

# Structure template

# Langages de programmation pour internet

## HTML, CSS, Javascript et PHP



# HTML5

## HTML5

La création d'un site internet nécessite du contenu. Pour définir celui-ci, un langage, HTML pour Hyper Text Markup Language ou langage de balises hypertexte, est nécessaire. HTML est donc un langage de balisage du contenu, on dit aussi du fond, d'un site.

Aujourd'hui, la norme *HTML5* est en vigueur. C'est donc sur celle-ci qu'on va se baser. Il s'agit d'une norme où les balises ont du sens, c'est-à-dire qu'elles signifient quelque chose, contrairement aux balises d'HTML4. On parlera de la sémantique d'HTML5.

# Structure de balisage

Attention, HTML 5 est moins strict que XHTML !  
Bonne pratique : fermer les balises.

## Ouverture - fermeture

```
<nom_balise> texte </nom_balise>
```

## Hiérarchie

```
<balise1><balise2> texte </balise2></balise1>
```

# Balilage

**Structure :** <html>, <head>, <body>

**Head :** <title>, <meta>, <style>, <link>, <script>

**Body :** <div>, <header>, <article>, <nav>, <aside>, <footer>, <h1>, <h2-5>, <p>, <a> (lien internet), <table>, <tr> (ligne), <td> (colonne), <hr /> (ligne), <img />, <audio />, <video />, <br /> (saut ligne), <b>, <strong>, <i>, <em>, <span>, <button>, ...

# Attributs

Les balises peuvent avoir des attributs ou propriétés :

```
<balise attribut1="valeur1" attribut2="valeur2"> ... </balise>
```

Exemples :

**Liens internet :**

```
<a href="http://www.wikipedia.org"> Wikipedia</a>
```

**Ligne horizontale :**

```
<hr width='150px' align='left' />
```

**Image :**

```

```

**Image - lien :**

```
<a href="http://www.google.ch">Google</a>
```

# Structure générale

La structure générale d'une page HTML est la suivante :

<code>&lt;!DOCTYPE html&gt;</code>	-> la déclaration du type HTML5 du document
<code>&lt;html&gt;</code>	-> balise d'ouverture du document
<code>&lt;head&gt;</code>	-> balise d'ouverture de l'entête
...	...
<code>&lt;/head&gt;</code>	-> balise de fermeture de l'entête
<code>&lt;body&gt;</code>	-> balise d'ouverture du corps du document
...	...
<code>&lt;/body&gt;</code>	-> balise de fermeture du corps du document
<code>&lt;/html&gt;</code>	-> balise de fermeture du document

# Structure générale

La structure générale de l'entête est la suivante :

<code>&lt;head&gt;</code>	-> balise d'ouverture de l'entête
<code>&lt;meta charset="UTF-8" /&gt;</code>	-> Codage des caractères utilisés
<code>&lt;meta name="description" content="Description de la page" /&gt;</code>	-> balise de description de la page
...	...
<code>&lt;meta name="author" content="mon nom" /&gt;</code>	-> balise de déclaration de l'auteur
...	...
<code>&lt;title&gt;Titre de la fenetre&lt;/title&gt;</code>	-> balise de titre de la fenêtre
<code>&lt;link rel="stylesheet" href="style.css" /&gt;</code>	-> balise de feuille de style externe
...	...
<code>&lt;style&gt; ... &lt;/style&gt;</code>	-> balise de feuille de style interne
<code>&lt;script&gt; ... &lt;/script&gt;</code>	-> balise de script
<code>&lt;/head&gt;</code>	-> balise de fermeture de l'entête

# Structure générale

La structure générale du corps est la suivante :

```
<body>  
  <header> ... </header>  
  <nav> ... </nav>  
  <aside> ... </aside>  
  <article> ... </article>  
  ...  
  <footer> ... </footer>  
</body>
```

- > balise d'ouverture du corps
- > balise de l'entête (logo, titre, ...)
- > balise de la barre de navigation
- > balise de colonne latérale droite
- > balises des articles
- > ...
- > balise du pied de page (copyright, ...)
- > balise de fermeture du corps

# CSS3

Les feuilles de style permettent de gérer la forme qu'on veut donner au contenu d'un site. Cette forme peut dépendre des appareils (smartphone, écrans, beamer, ...) utilisés. Pour la gérer au mieux, on utilise généralement une feuille de style par appareil.

Un langage, CSS pour Cascading Style Sheet ou feuilles de style en cascade, est nécessaire pour cela. Aujourd'hui, la norme CSS3 est en vigueur. C'est donc sur celle-ci qu'on va se baser.



# Intégration des feuilles de styles

Trois cas sont envisageables :

Dans la balise.

```
<p style="color:red;font-size:100">Texte</p>
```

Feuille interne

```
<head>  
  <style type="text/css">  
  
  </style>  
</head>
```

Feuille externe

```
<head>  
<link rel="stylesheet"  
      type="text/css" href="  
      feuille.css" />  
</head>
```

# Feuilles de styles : CSS

Idée : séparer le contenu de la forme. Syntaxe :

```
sélecteur {propriété :valeur}
```

## Exemples :

```
h1 {font-size :120;}
```

```
p {font-family :“sans serif” ;  
font-size :90% ; color :red}
```

## Meilleure indentation :

```
h1, h2 {
```

```
font-family : “sans serif” ;  
color : red ;
```

```
}
```

# Feuilles de styles : sélecteurs

## Sélecteurs imbriqués :

Toutes les sélections (span) de chaque paragraphes (p) sont passées en rouge.

```
<p>La voiture est  
<span>très</span> petite</p>
```

```
p span {color : red ;}
```

La voiture est très petite

## Sélecteur identifié :

Attention le sélecteur identifié est unique ! En d'autres termes, on ne peut mettre plusieurs identifiants identiques.

```
<p id="cetexte">Un pti texte</p>
```

```
#cetexte {color : red}
```

Un ptit texte

# Feuilles de styles : CSS

**Sélecteurs de classe** : Il s'agit d'appliquer un style à un ensemble d'éléments donnés.

```
<h1 class="laiclass">La </h1>  
<em class="laiclass">deux chevaux,  
</em>  
<p class="laiclass">sa puissance  
</p>  
<p>c'est la lenteur.</p>  
.laiclass {color :red}
```

**La deux chevaux, sa puissance**  
c'est la lenteur.

**Sélecteur contextuel** : Il s'agit d'appliquer un style suivant le contexte.

```
<div>Le vélo, c'est bien</div>  
div :hover {background-color : red ;}  
Ici, en passant la souris sur le div, il  
devient rouge.
```

# Feuilles de styles : CSS

**Plusieurs sélecteurs** : Il s'agit d'appliquer le même style à plusieurs sélecteurs différents.

```
<h1>La </h1>  
<em>deux chevaux, </em>  
<p>sa puissance </p>  
<p>c'est la lenteur.</p>
```

```
h1, em, p {color :red}
```

**La** *deux chevaux***, sa puissance**  
**c'est la lenteur.**

**Et d'autre encore ... A découvrir.**

# Javascript

## Javascript

Javascript est un langage de programmation complet du côté client, c'est-à-dire que c'est le navigateur qui va l'utiliser pour enrichir les pages web d'une interactivité manquante à HTML et CSS.

L'utilisation de Javascript peut être problématique pour deux raisons. Premièrement, tournant du côté client, il ne permet pas l'accès aux bases de données se trouvant sur le serveur, par exemple. Deuxièmement, il peut avoir accès au client et est donc potentiellement dangereux.

# Bonnes pratiques

## Appel de JS

Pour éviter de bloquer le site si JS n'est pas activé, il faut le charger en dernier. On parle de JS non intrusif.

L'instruction suivante est à placer juste après la fermeture de la balise body :

```
</body>  
<script src='js/monjs.js'></script>
```

## Séparation du JS

Comme pour CSS, il faut placer le code JS dans un fichier séparé du HTML.

Généralement, on le place dans le répertoire js/ avec pour nom monjs.js. Attention, l'extension .js est impérative.

# Animation javascript 1 : image, gif animé

## Première idée :

```
<img src='tux.png' />
```

## ... avec un gif animé ...

```
<img src='robot.gif' />
```

## Déplacement du gif animé en javascript

```
function deplace(){  
  identification = document.getElementById('robot')  
  position = parseInt(identification.style.left)  
  position = position + 10  
  identification.style.left = position+"px"  
}
```

Résultat

Code



# Animation javascript 2 : tableau d'images

On crée un tableau de références d'images :

## Tableau

```
var imgFiles = new Array (  
  "Cours/Cours_e0000.gif",  
  "Cours/Cours_e0001.gif",  
  "Cours/Cours_e0002.gif",  
  "Cours/Cours_e0003.gif",  
  "Cours/Cours_e0004.gif",  
  "Cours/Cours_e0005.gif",  
  "Cours/Cours_e0006.gif",  
  "Cours/Cours_e0007.gif"  
)
```

On appelle à intervalle régulier la fonction :

## Animation

```
function animate(){  
  frame += 1  
  if (frame > imgFiles.length){  
    frame = 0  
  }  
  spritelImage.src = imgFiles[frame]  
}
```

Résultat

Code

# Animation javascript 3 : une seule image

## Frames juxtaposées en css

```
#contenant {  
  background-image: url("Cours/Cours_8  
  .gif");  
  height: 128px;  
  width: 128px;  
  background-position: 0px 0px;}
```

## Tableau des décalages

```
var offsetList = new Array(0, -128,  
  -256, -384, -512, -640, -768,  
  -896)
```

## Parcours de l'image

```
function cours(){  
  frame++  
  if (frame >= offsetList.length){  
    frame = 0  
  }  
  offset = offsetList[frame] + "px 0px  
  "  
  contenant.style.backgroundColor =  
    offset  
}
```

[Résultat](#)[Code](#)

# Animation javascript 4 : le déplacement

## Gestion du clavier

```
document.onkeypress = keyListener
```

## Détection des touches

```
function keyListener(e){  
  if (e.keyCode == 37){  
    moveContenant(-10, 0)  
  } // gauche  
  if (e.keyCode == 38){  
    moveContenant(0, -10)  
  } // haut ... }  
}
```

## Déplacement de l'image

```
function moveContenant(dx, dy){  
  x = parseInt(contenant.style.left)  
  y = parseInt(contenant.style.top)  
  x += dx  
  y += dy  
  contenant.style.left = x + "px"  
  contenant.style.top = y + "px"  
}
```

Résultat

Code

# Animation javascript 5 : le tout

Un joli résultat

Résultat

Un joli "petit" programme

Code

# Crédits photographiques I

L'ensemble des illustrations utilisées dans cette présentation sont publiées sur le net soit sous licence GFDL, soit dans le domaine public. Ci-dessous se trouvent les liens permettant de vérifier les licences. Ils donnent aussi accès aux auteurs que je remercie tout particulièrement pour leur travail sans lequel cette présentation n'aurait pas pu exister.

<sup>1</sup>Wikipedia : <http://commons.wikimedia.org/wiki/File:Notes.svg>

<sup>2</sup>Wikipedia : [http://commons.wikimedia.org/wiki/File:Phoenix\\_on\\_the\\_Red\\_Planet.jpg](http://commons.wikimedia.org/wiki/File:Phoenix_on_the_Red_Planet.jpg)

<sup>3</sup>Wikipedia : [https://upload.wikimedia.org/wikipedia/commons/thumb/8/8b/Raspberry\\_Pi\\_4%2C\\_2\\_GB\\_RAM\\_version.jpg/800px-Raspberry\\_Pi\\_4%2C\\_2\\_GB\\_RAM\\_version.jpg?uselang=fr](https://upload.wikimedia.org/wikipedia/commons/thumb/8/8b/Raspberry_Pi_4%2C_2_GB_RAM_version.jpg/800px-Raspberry_Pi_4%2C_2_GB_RAM_version.jpg?uselang=fr)

<sup>4</sup>Wikimedia : [https://upload.wikimedia.org/wikipedia/commons/thumb/6/61/VGA\\_Stecker.jpg/274px-VGA\\_Stecker.jpg](https://upload.wikimedia.org/wikipedia/commons/thumb/6/61/VGA_Stecker.jpg/274px-VGA_Stecker.jpg)

<sup>5</sup>Wikimedia : <https://upload.wikimedia.org/wikipedia/commons/thumb/3/37/Dvi-cable.jpg/300px-Dvi-cable.jpg>

<sup>6</sup>Wikimedia : <https://commons.wikimedia.org/wiki/File:HDMI.jpg>

<sup>7</sup>Wikimedia : [https://upload.wikimedia.org/wikipedia/commons/thumb/f/f4/USB\\_types\\_2.jpg/320px-USB\\_types\\_2.jpg](https://upload.wikimedia.org/wikipedia/commons/thumb/f/f4/USB_types_2.jpg/320px-USB_types_2.jpg)

<sup>8</sup>Wikimedia : [https://upload.wikimedia.org/wikipedia/commons/thumb/c/c1/Usb\\_type-c\\_plug\\_socket.jpg/320px-Usb\\_type-c\\_plug\\_socket.jpg](https://upload.wikimedia.org/wikipedia/commons/thumb/c/c1/Usb_type-c_plug_socket.jpg/320px-Usb_type-c_plug_socket.jpg)

<sup>9</sup>Wikimedia : <https://upload.wikimedia.org/wikipedia/commons/thumb/3/39/RJ45top.jpg/320px-RJ45top.jpg>

<sup>10</sup>Voir wikimedia commons : [https://commons.wikimedia.org/wiki/File:Fairphone\\_3\\_modules\\_on\\_display.jpg](https://commons.wikimedia.org/wiki/File:Fairphone_3_modules_on_display.jpg)

<sup>11</sup>Wikipedia : [http://fr.wikipedia.org/wiki/Logiciels\\_libres](http://fr.wikipedia.org/wiki/Logiciels_libres)

<sup>12</sup>Wikimedia : [https://commons.wikimedia.org/wiki/File:Louvre\\_plan.png?uselang=fr](https://commons.wikimedia.org/wiki/File:Louvre_plan.png?uselang=fr)

<sup>13</sup>Wikimedia : [https://commons.wikimedia.org/wiki/File:Reseau\\_arborescent.svg?uselang=fr](https://commons.wikimedia.org/wiki/File:Reseau_arborescent.svg?uselang=fr)

<sup>14</sup>Voir Wikimedia commons : [https://commons.wikimedia.org/wiki/File:Chiffrement\\_asymétrique.jpg](https://commons.wikimedia.org/wiki/File:Chiffrement_asymétrique.jpg)

<sup>15</sup>Voir le mode d'emploi « Autodéfense courriel » : <https://emailselfdefense.fsf.org/fr/>

# Crédits photographiques II

<sup>16</sup>Voir Wikimedia commons : [https://commons.wikimedia.org/wiki/File:Computer\\_tapes.jpg](https://commons.wikimedia.org/wiki/File:Computer_tapes.jpg)

<sup>17</sup>Voir Wikimedia commons : [https://commons.wikimedia.org/wiki/File:Disque\\_dur\\_0004.JPG](https://commons.wikimedia.org/wiki/File:Disque_dur_0004.JPG)

<sup>18</sup>Voir Wikimedia commons : [https://commons.wikimedia.org/wiki/File:Vertex\\_2\\_Solid\\_State\\_Drive\\_by\\_OCZ-top\\_oblique\\_PNr%C2%B00307.jpg](https://commons.wikimedia.org/wiki/File:Vertex_2_Solid_State_Drive_by_OCZ-top_oblique_PNr%C2%B00307.jpg)

<sup>19</sup>Voir Wikimedia commons : [https://commons.wikimedia.org/wiki/File:16\\_USB\\_flash\\_drives.jpg](https://commons.wikimedia.org/wiki/File:16_USB_flash_drives.jpg)

<sup>20</sup>Voir Wikimedia commons : [https://commons.wikimedia.org/wiki/File:Bildverarbeitung\\_blob\\_sdcard.png](https://commons.wikimedia.org/wiki/File:Bildverarbeitung_blob_sdcard.png)

png

<sup>21</sup>Voir wikimedia commons : [https://commons.wikimedia.org/wiki/File:Edward\\_Snowden.jpg](https://commons.wikimedia.org/wiki/File:Edward_Snowden.jpg)

<sup>22</sup>Wikipedia : <http://commons.wikimedia.org/wiki/File:Notes.svg>

<sup>23</sup>Voir wikipedia : [http://commons.wikimedia.org/wiki/File:CCD\\_in\\_camera.jpg](http://commons.wikimedia.org/wiki/File:CCD_in_camera.jpg)

<sup>24</sup>Voir wikipedia : <http://commons.wikimedia.org/wiki/File:Pixel.jpg>

<sup>25</sup>Voir wikipedia : [http://fr.wikipedia.org/wiki/Fichier:Resolution\\_test.jpg](http://fr.wikipedia.org/wiki/Fichier:Resolution_test.jpg)

<sup>26</sup>Voir wikipedia : <http://commons.wikimedia.org/wiki/File:BayerPatternFiltration.png>

<sup>27</sup>Voir Wikicommon : [http://commons.wikimedia.org/wiki/File:Fourier\\_d%27un\\_carr%C3%A9.svg](http://commons.wikimedia.org/wiki/File:Fourier_d%27un_carr%C3%A9.svg)

<sup>28</sup>Voir Wikicommon : [http://commons.wikimedia.org/wiki/File:Onde\\_endroit\\_fixe.png](http://commons.wikimedia.org/wiki/File:Onde_endroit_fixe.png)