Option complémentaire d'informatique

Lycée Blaise Cendrars

17 septembre 2020

Table des matières

Interdisciplinarité

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

Option complémentaire

- 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[1].

En bas :

Aspirateur-récureur iRobot©.

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.
- Pérénité 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.
- Pérénité d'un groupe de programmeurs.

Difficultés

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

Évolution

Informatique

Le lièvre et la tortue ou le renard et le hérisson ...

Lièvre ou tortue

Évolution technique très rapide

$\downdownarrows \upuparrows$

Besoins réels stables

Renard ou hérisson

Évolution inexorable

$\downdownarrows \upuparrows$

Stabilité nécessaire

Contexte de l'Option Complémentaire

1. - Genève : Logiciels libres.
   - Vaud : MacOS©.
   - Neuchatel : Windows©.
2. Crise économique mondiale et cantonale $\Rightarrow$ Crédits zéro en 2010.
3. Compétences présentes en informatiques au lycée : LINUX ET MACOS©.
4. Contexte éducatif et coopératif.

$\Leftrightarrow$

Contexte de l'Option Complémentaire

1. - Genève : Logiciels libres.
   - Vaud : MacOS©.
   - Neuchatel : Windows©.
2. Crise économique mondiale et cantonale $\Rightarrow$ Crédits zéro en 2010.
3. Compétences présentes en informatiques au lycée : LINUX ET MACOS©.
4. Contexte éducatif et coopératif.

$\Leftrightarrow$

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.

Table des matières

Systèmes d'exploitations

OS : operating system

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

Principaux OS

Windows	MacOS©	Linux©
Installé par défaut ou acheté en pack	Installé par défaut sur machine Apple	De type Unix installable "partout"
Payant	Payant	Gratuit

Systèmes d'exploitation propriétaires

Mac OS©

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, openoffice, gimp, . . .

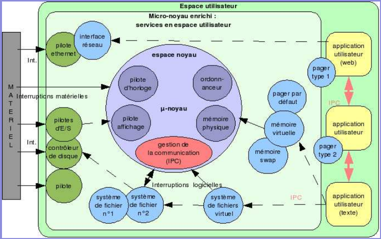
Windows©

Windows©se décline en XP, Seven et 8, orientée pc-tablettes.
XP aurait du être abandonné, mais il a été récupéré pour investir le domaine des ultraportables et pour palier au peu de succès de Vista qui n'a pas été la réussite attendue.

Architectures

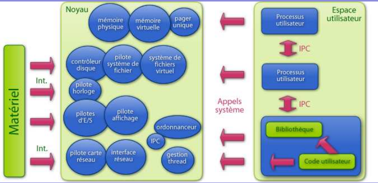
L'ensemble des logiciels d'un ordinateur se décompose dans deux espaces : l'espace noyau – l'espace utilisateur

Micronoyau



Noyau des système Windows©et MacOSX©[2].

Noyau monolitique



Noyau du système Linux[3].

Table des matières

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 Creatives Commons. Relevons aussi que Windows©et MacOS©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[4].

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 monolitique et d'un ensemble de logiciels libres.

Distributions Linux©

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.

Table des matières

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, ...

Table des matières

Composants

Un ordinateur est construit autour des composants suivants :

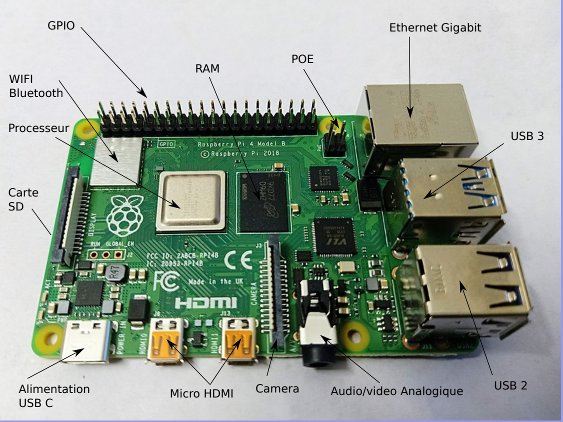
Processeur Un circuit électronique qui réalise les calculs,

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



Référence image[5]

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'ether : protocole de communication local.

Table des matières

Table des matières

Table des matières

Texte et codage

Caractères ⇆ 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 carcatère de l'euro.

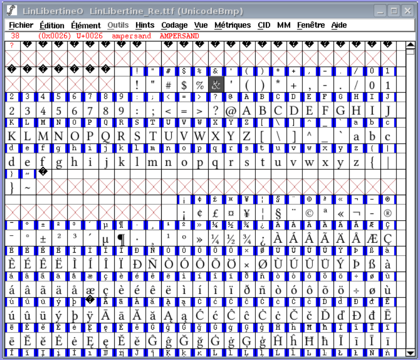
Norme UTF (Universal coded character set Transformation Format)

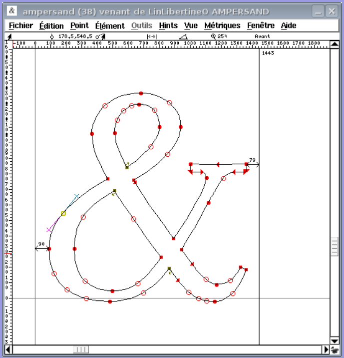
Table de codage unique : "Unicode".

Elle sépare la représentation du caractère du codage lui-même par l'intermédiaire d'un index numérique nomé *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.





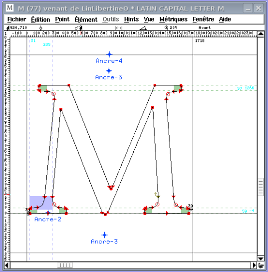
Fontes – bésier – bitmap

Fontes vectorielles – Fontes bitmap

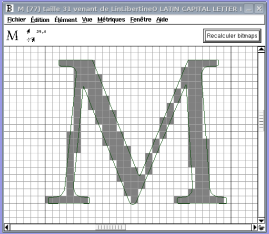
Postscript, truetype – bdf

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

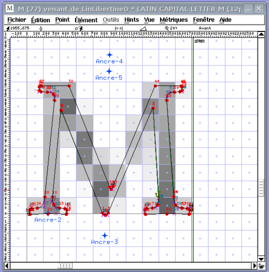
Courbe de bésier



Représentation bitmap



Anti-aliasing et hinting



Autres problèmes rencontrés : gestion des crénages (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 glyphes :
avec ou sans empattement – avec ou sans serif – à taille fixe ou pas

AaBbCcFcTe AaBbCcFcTe AaBbCcFcTe

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©, OpenOffice, 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.

Scribus

Domaine : arts graphiques, affiches, revues, journaux.

Idée : aide au positionnement des images, répartition du texte dans des blocs dépendants.

Geleneksel Türk Sanatı : Ebru

Ebru'nun Türk ekincinde yeri ilen günayı...

Ortaya çıkış yeri ve tarihine ilişkin kesin bir delil bulunmamaktadır. Ancak, köklerinin 9. ve 10. yüzyıla kadar uzandığı varsayılmaktadır. Bilinen o ki, bu sanat, kağıdın tarih sahnesine girmesiyle gelişmiştir. Çin'de liu-şa-cien, XII. asırdan itibaren Japonya'da suminagaşi ve beninagaşi isimleriyle sulu vasatta yapılan bir takım çalışmaların mevcudiyeti, daha sonraki asırlarda Çağatay Türkçesi'yle ebre adını alarak Türkistan'da ortaya çıkan bu sanatın tarihi gelişimi hakkında, müphem de olsa bir fikir vermektedir. Türkistan'dan en geç XVI. asır başlarında İpekyolu'nu takiben İran'a geçişinde ebri olarak isimlendirilen bu sanat, görünüşüyle gerçekten bulut kümelerine benzer şekiller taşıdığından, buluta nisbet ifade eden bu Farsça ismi doğrulamaktadır. Osmanlı ülkesinde de revaç bulan aynı isim, telaffuz zorluğundan son yüzyılda Türkçe'de ebru'ya dönüşmüştür. Galat olmakla beraber, kaş gibi şekiller de ihtiva ettiğinden, bu sanata ebru denilmesi bir çelişki sayılmamalıdır; çünkü ebru kelimesi Farsça'da kaş manasına gelmektedir. XVI. asır ortalarında Mir Muhammed Tahir tarafından Hindistan'da yapılmaya başlandığı rivayet olunan ebruculuk, buradan İran'a ve sonra da İstanbul'a kadar yayılmıştır. Aynı yüzyılın sonlarında, İstanbul'dan Avrupalı seyyahlar tarafından kendi memleketlerine götürülen ebru kağıtları önce Almanya'da, sonra da Fransa ve İtalya'da mermer kağıdı veya Türk mermer kağıdı, hatta sadece Türk kağıdı adıyla tanınıp benimsenmiş ve oralarda da yapılmaya başlanmıştır. Zaman içinde İngiltere ve Amerika'ya da yayılan ebru kağıdı, her ülkenin sanat anlayışına göre bir başkalık gösterir. Bunda, kullanılan değişik malzemenin de rolü olmalıdır. Belgelenen en eski ebru örneği 16. yüzyıla aittir. Kağıdın süslenmesinde, kıt'a ve levhaların iç ve dış pervazlarında, yazma ciltlerinde yan kağıdı olarak sıkça kullanılmıştır. (Derman M.Uğur Osmanlı Ansiklopedisi. C.11,s.189)

Ebru hakkında Türkçe kaleme alınmış bilinen en eski eser, 1615'ten sonra yazılan "Tertib-i Risâle-i Ebrî" adlı yazma kitapçıktır. Günümüzde bilinen ebru tarzındaki eserler ilk kez Orta Asya - Osmanlı coğrafyasında ortaya çıkmıştır. Ebrunun tarihi ile ilgili olarak sayın Uğur Derman (Türk Santında Ebrû), tarihi kestirilebilen en eski ebru olarak, üzerinde Mâlikî Deylemî'ye ait bir kıt'anın bulunduğu ve Gürcistan'da yazılmış olan 1554 tarihli bir ebruyu gösterir. Bu ebrunun, hafif ebru olarak yapılmış olması ve hafif ebrunun ancak belli bir ustalaşmadan sonra yapılabildiği gözönüne alınacak olursa, ebrunun orijinin çok daha eskilere dayandığı düşünülmektedir.

Geleneksel Türk sanatlarından Ebru; yoğunlaştırılmış su üzerine betiz yapma sanatıdır.

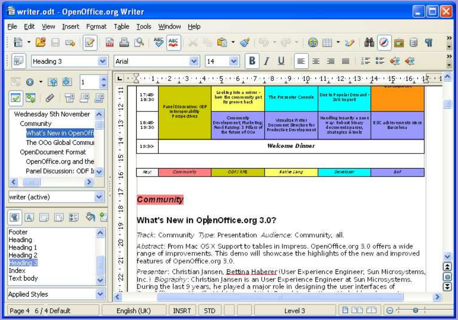
Osmanlı döneminde başlıbaşına bir sanat ve iş kolu olan ebruculuk, 20.yüzyıl başlarına gelindiğinde unutulma noktasına gelmiştir. Bu sanatın tekrar hayat kazanması, ebru sanatına 'çiçekli ebru'yu hediye eden büyük sanatçı Necmeddin Okyay sayesinde olmuştur. Okyay'dan sonraki büyük merhale Mustafa Düzgünman'dır. Mustafa Düzgünman'ın talebelerinden Niyazi Sayın, Sabri Mandıracı,Timuçin Tanarslan,Alparslan Babaoğlu,Fuat Başar bu sanatın yeniden yaygınlaşmasında büyük gayret sarfetmişlerdir. ebru sanatı nın tanınması ebru sanatı birçok dizide filmde ve tanıtımlarda kullanılmıştır.bu yüzlen ülkemizde herkes ebru sanatını az da olsa bilir.

Kaynak : Vikipedi

Bilmem Ne Dergisi - Mayıs Sayısı 2008 www.gokbey-uluc.blogspot.com

OpenOffice

Traitement de texte classique. Bien que ...

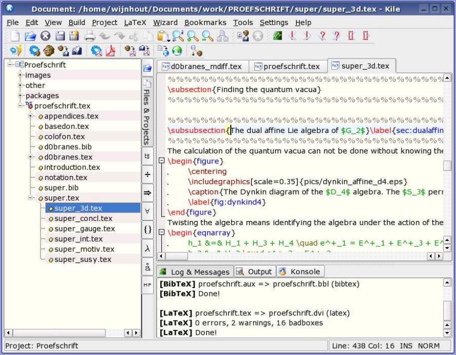


Gestion avancée des styles, références, bibliographie, index, ...

OpenOffice avec styles

Latex

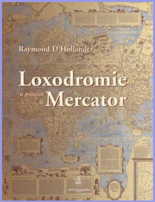
Traitement de texte haute qualité :
gestion professionnelle automatique de la typographie.

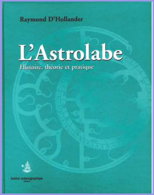


Exemples :

"L'astrolabe, histoire théorie et pratique" et

"Loxodromie et projection de Mercator".



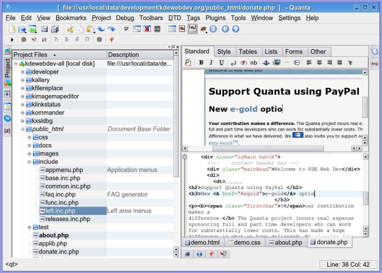


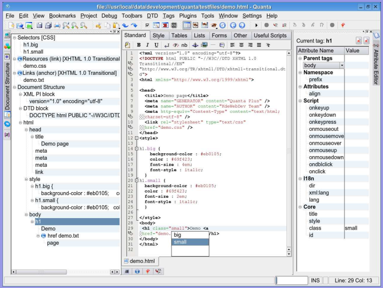
Autre exemple connu : votre"formulaire et tables"...

Xhtml et Css

Orientation web – séparation du contenu et de la forme.

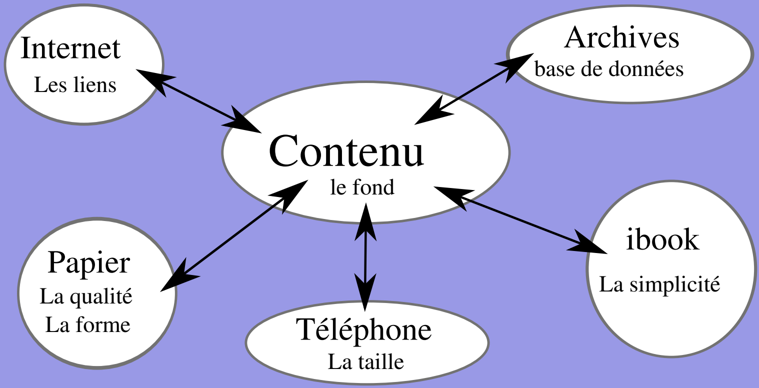
Langages balisés sans boucles, fortement normalisés et strict.





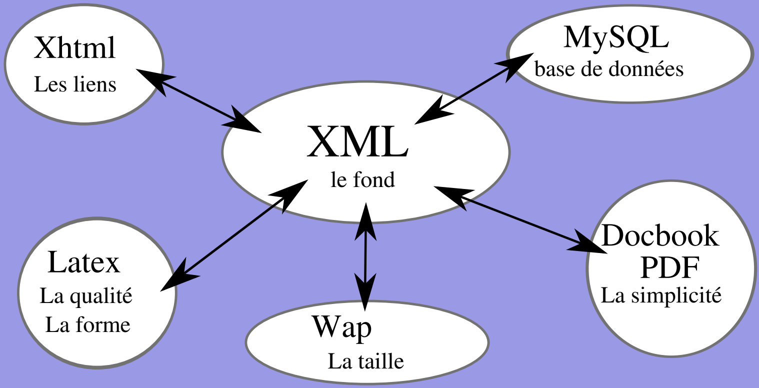
Publication multi-support

Aujourd'hui les supports d'édition sont multiples.



Publication multi-support

Et les langages nécessaires aussi (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 language 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>
```

Évaluations

Désagréables $\longleftrightarrow$ agréables[6] ?

Description des évaluations

Table des matières

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 :

$$127 = 100+20+7 = 1\cdot100+2\cdot10+7\cdot1 =$$

Représentation d'un nombre

Comment un ordinateur comprend-il les nombres ?

Représentation décimale :

$$127 = 100+20+7 = 1\cdot100+2\cdot10+7\cdot1 = 1\cdot10^2+2\cdot10^1+7\cdot10^0$$

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 :

$$127 = 100+20+7 = 1\cdot100+2\cdot10+7\cdot1 = 1\cdot10^2+2\cdot10^1+7\cdot10^0$$

Représentation binaire :

$$1101 =$$

Représentation d'un nombre

Comment un ordinateur comprend-il les nombres ?

Représentation décimale :

$$127 = 100+20+7 = 1\cdot100+2\cdot10+7\cdot1 = 1\cdot10^2+2\cdot10^1+7\cdot10^0$$

Représentation binaire :

$$1101 = 1\cdot2^3+1\cdot2^2+0\cdot2^1+1\cdot2^0 =$$

Représentation d'un nombre

Comment un ordinateur comprend-il les nombres ?

Représentation décimale :

$$127 = 100+20+7 = 1\cdot100+2\cdot10+7\cdot1 = 1\cdot10^2+2\cdot10^1+7\cdot10^0$$

Représentation binaire :

$$1101 = 1\cdot2^3+1\cdot2^2+0\cdot2^1+1\cdot2^0 = 8+4+0+1 = 13$$

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 $1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} = 14{,}25$

Nombres à virgule flottante

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

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

avec $1 \leq mantisse < 10$
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 :

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

avec $1 \leq mantisse < 2$
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).

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

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

Exemple numérique

$$\begin{aligned} Nombre &= 10101000000000000000000000000110 \\ &= 1 \,|\, 01010000000000000000000 \,|\, 00000110 \\ &= signe \,|\, mantisse \,|\, exposant \end{aligned}$$

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

- Le signe $= 1 \Rightarrow$ nombre négatif.
- La mantisse $= 1 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} + 0 \cdot 2^{-3} + 1 \cdot 2^{-4} = 1,3125$
- 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.

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

Exercices

Convertissez les nombres suivants :

- 00110000000000000000000010000000 en décimal
- 00111000000000000000000011000000 en décimal
- 00101000000000000000000010100000 en décimal
- 10110000000000000000000001010000 en décimal

Exercices

Convertissez les nombres suivants :

- 0 | 01100000000000000000000 | 10000000
- 00111000000000000000000011000000 en décimal
- 00101000000000000000000010100000 en décimal
- 10110000000000000000000001010000 en décimal

Exercices

Convertissez les nombres suivants :

- $0 \,|\, 01100000000000000000000 \,|\, 10000000$
  $+ \,\Big|\, 1 + 2^{-2} + 2^{-3} = 1,375 \,\Big|\, 2^7 - 127 = 1 \;\Rightarrow$
- 00111000000000000000000011000000 en décimal
- 00101000000000000000000010100000 en décimal
- 10111000000000000000000001010000 en décimal

Exercices

Convertissez les nombres suivants :

- $0 \,|\, 01100000000000000000000 \,|\, 10000000$
  $+ \,|\, 1 + 2^{-2} + 2^{-3} = 1,375 \,|\, 2^{7} - 127 = 1 \;\Rightarrow$
  $+1,375 \cdot 2^{1} = 2,75$
- 00111000000000000000000011000000 en décimal
- 00101000000000000000000010100000 en décimal
- 10111000000000000000000001010000 en décimal

Exercices

Convertissez les nombres suivants :

- 00110000000000000000000010000000= $+1,375 \cdot 2^1 = 2,75$
- 00111000000000000000000011000000 en décimal
- 00101000000000000000000010100000 en décimal
- 10110000000000000000000001010000 en décimal

Exercices

Convertissez les nombres suivants :

- 00110000000000000000000010000000= $+1,375 \cdot 2^1 = 2,75$
- 0 | 0111000000000000000000000 | 11000000
- 00101000000000000000000010100000 en décimal
- 10110000000000000000000001010000 en décimal

Exercices

Convertissez les nombres suivants :

- 00110000000000000000000010000000=
  $+1,375 \cdot 2^1 = 2,75$
- $0 \mid 01110000000000000000000 \mid 11000000$
  $+ \mid 1 + 2^{-2} + 2^{-3} + 2^{-4} = 1,4375 \mid 2^7 + 2^6 - 127 = 65 \Rightarrow$
- 00101000000000000000000010100000 en décimal
- 10110000000000000000000001010000 en décimal

Exercices

Convertissez les nombres suivants :

- 00110000000000000000000010000000=
  $+1,375 \cdot 2^{1} = 2,75$
- $0 \,|\, 01110000000000000000000 \,|\, 11000000$
  $+ \,|\, 1 + 2^{-2} + 2^{-3} + 2^{-4} = 1,4375 \,|\, 2^{7} + 2^{6} - 127 = 65 \Rightarrow$
  $+1,4375 \cdot 2^{65} = 5,3 \cdot 10^{19}$
- 00101000000000000000000010100000 en décimal
- 10110000000000000000000001010000 en décimal

Exercices

Convertissez les nombres suivants :

- 00110000000000000000000010000000=
  $+1,375 \cdot 2^1 = 2,75$
- 00111000000000000000000011000000=
  $+1,4375 \cdot 2^{65} = 5,3 \cdot 10^{19}$
- 00101000000000000000000010100000 en décimal
- 10110000000000000000000001010000 en décimal

Exercices

Convertissez les nombres suivants :

- 00110000000000000000000010000000= $+1,375 \cdot 2^1 = 2,75$
- 00111000000000000000000011000000= $+1,4375 \cdot 2^{65} = 5,3 \cdot 10^{19}$
- 0 | 01010000000000000000000 | 10100000
- 10110000000000000000000001010000 en décimal

Exercices

Convertissez les nombres suivants :

- 00110000000000000000000010000000=
  $+1,375 \cdot 2^1 = 2,75$
- 00111000000000000000000011000000=
  $+1,4375 \cdot 2^{65} = 5,3 \cdot 10^{19}$
- $0 \mid 01010000000000000000000 \mid 10100000$
  $+ \mid 1 + 2^{-2} + 2^{-4} = 1,3125 \mid 2^7 + 2^5 - 127 = 33 \; \Rightarrow$
- 10110000000000000000000001010000 en décimal

Exercices

Convertissez les nombres suivants :

- 00110000000000000000000010000000=
  $+1,375 \cdot 2^1 = 2,75$
- 00111000000000000000000011000000=
  $+1,4375 \cdot 2^{65} = 5,3 \cdot 10^{19}$
- $0 \,\big|\, 01010000000000000000000 \,\big|\, 10100000$
  $+ \,\big|\, 1 + 2^{-2} + 2^{-4} = 1,3125 \,\big|\, 2^7 + 2^5 - 127 = 33 \Rightarrow$
  $+1,3125 \cdot 2^{33} = 1,1 \cdot 10^{10}$
- 10110000000000000000000010100000 en décimal

Exercices

Convertissez les nombres suivants :

- 00110000000000000000000010000000= $+1,375 \cdot 2^1 = 2,75$
- 00111000000000000000000011000000= $+1,4375 \cdot 2^{65} = 5,3 \cdot 10^{19}$
- 00101000000000000000000010100000= $+1,3125 \cdot 2^{33} = 1,1 \cdot 10^{10}$
- 10110000000000000000000001010000 en décimal

Exercices

Convertissez les nombres suivants :

- 00110000000000000000000010000000=
  $+1,375 \cdot 2^{1} = 2,75$
- 00111000000000000000000011000000=
  $+1,4375 \cdot 2^{65} = 5,3 \cdot 10^{19}$
- 00101000000000000000000010100000=
  $+1,3125 \cdot 2^{33} = 1,1 \cdot 10^{10}$
- 1 | 01100000000000000000000 | 01010000

Exercices

Convertissez les nombres suivants :

- 00110000000000000000000010000000=
  $+1,375 \cdot 2^{1} = 2,75$
- 00111000000000000000000011000000=
  $+1,4375 \cdot 2^{65} = 5,3 \cdot 10^{19}$
- 00101000000000000000000010100000=
  $+1,3125 \cdot 2^{33} = 1,1 \cdot 10^{10}$
- $1 \,\Big|\, 01100000000000000000000 \,\Big|\, 01010000$
  $- \,\Big|\, 1 + 2^{-2} + 2^{-3} = 1,375 \,\Big|\, 2^{6} + 2^{4} - 127 = -47 \;\Rightarrow$

Exercices

Convertissez les nombres suivants :

- 00110000000000000000000010000000=
  $+1,375 \cdot 2^{1} = 2,75$
- 00111000000000000000000011000000=
  $+1,4375 \cdot 2^{65} = 5,3 \cdot 10^{19}$
- 00101000000000000000000010100000=
  $+1,3125 \cdot 2^{33} = 1,1 \cdot 10^{10}$
- $1 \mid 01100000000000000000000 \mid 01010000$
  $- \mid 1 + 2^{-2} + 2^{-3} = 1,375 \mid 2^{6} + 2^{4} - 127 = -47 \Rightarrow$
  $-1,375 \cdot 2^{-47} = -9,8 \cdot 10^{-15}$

Exercices

Convertissez les nombres suivants :

- 00110000000000000000000010000000= $+1,375 \cdot 2^{1} = 2,75$
- 00111000000000000000000011000000= $+1,4375 \cdot 2^{65} = 5,3 \cdot 10^{19}$
- 00101000000000000000000010100000= $+1,3125 \cdot 2^{33} = 1,1 \cdot 10^{10}$
- 10110000000000000000000001010000= $-1,375 \cdot 2^{-47} = -9,8 \cdot 10^{-15}$

Table des matières

Byte, bit et octet

Décimal

$0, 1, 2, \ldots, 9$: chiffres.

$23, 45, 234, \ldots$: nombres.

Byte, bit et octet

Décimal

$0, 1, 2, \ldots, 9$: chiffres.

$23, 45, 234, \ldots$: nombres.

Binaire

0 et 1 : **bits.**

$101, 11, 10, \ldots$: nombres.

Byte, bit et octet

Décimal

$0, 1, 2, \ldots, 9$: chiffres.

$23, 45, 234, \ldots$: nombres.

Binaire

0 et 1 : **bits.**

$101, 11, 10, \ldots$: nombres.

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

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

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 ≡ 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 ≡ nombre de possibilités de réaliser un nombre à l'aide de 3 chiffres :

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

Binaire

kilo ≡ 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 millons.
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

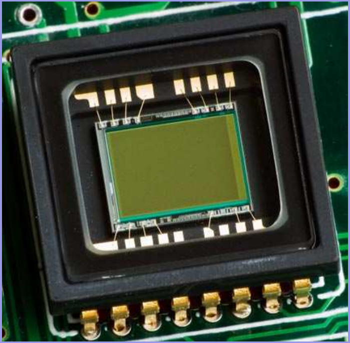
$$2^{127} = 1,7 \cdot 10^{38}$$

Au total, on a donc le nombre maximal

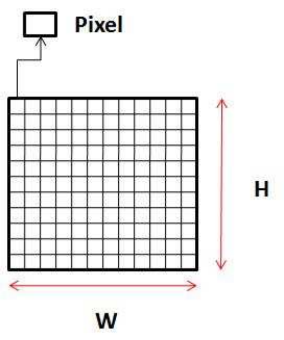
$$1,9999 \cdot 1,7 \cdot 10^{38} = 3,4 \cdot 10^{38}$$

Image matricielle ou bitmap

Capteur ccd[7] : niveaux de gris.



Grille de pixels[8]

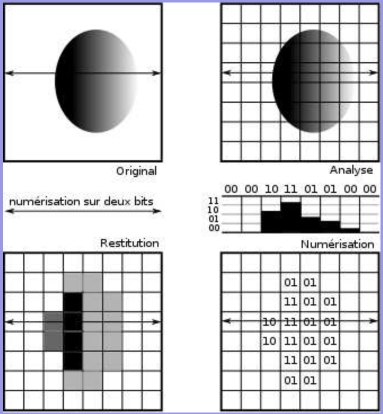


Numérisation sur deux bits

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

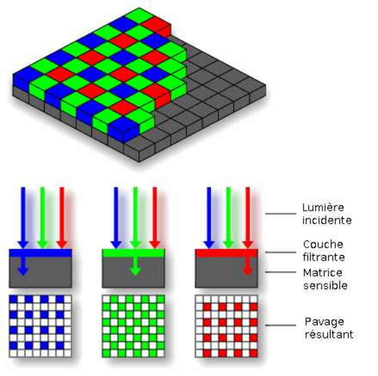
Unités de la résolution[9] : dpi (dot per inch) ou ppp (point par pouce).
Exemples typiques : écran 75-100 dpi, photocopieuse 360 dpi, imprimante 700 dpi, scanner 2000 dpi.

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).

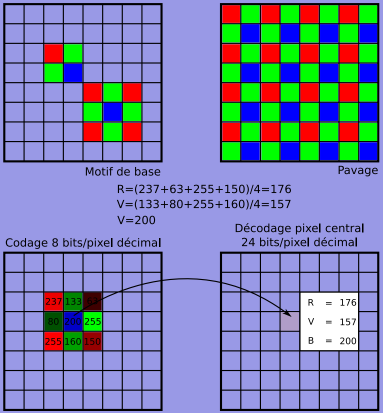
Norme : par la Commission Internationale de l'éclairage aux longueurs d'ondes de 546,1 nanomètre pour le rouge, 700,0 nanomètre pour le vert et 435,8 nanomètre pour le bleu.

Filtre de Bayer[10].



Filtre de Bayer

Filtre de Bayer.



- Pavage privilégiant le vert (œil plus sensible),
- Couleur pixel central :

  Bleu = 200

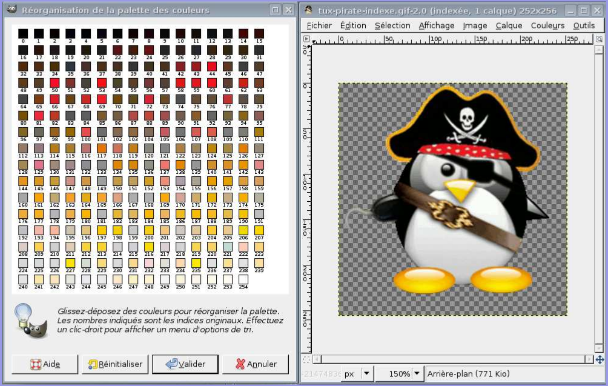
  Rouge = (237+63+255+150)/4=176

  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 x 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 x 256 pixels = 64′512 pixels

Couleurs vraies : 3 octets / pixel
3· 64′512 = 193′536 octets

193′536 octets = 189 ko

Total : **189 ko**
En réalité : 65,2 ko (compression)

Couleurs indexées

Image 252 x 256 pixels = 64′512 pixels

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

Palette 256 x 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

```
BMF.....
..6...(.
........
........
........
........
......ÿ.
.ÿÿÿ....
ÿ.ÿ...
```

Éditeur binaire

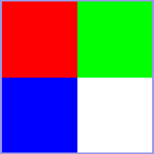
```
01000010 01001101 01000110 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00110110 00000000 00000000 00000000 00101000 00000000
00000000 00000000 00000010 00000000 00000000 00000000 00000010 00000000
00000000 00000000 00000001 00000000 00011000 00000000 00000000 00000000
00000000 00000000 00010000 00000000 00000000 00000000 00010011 00001011
00000000 00000000 00010011 00001011 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 11111111 00000000
00000000 11111111 11111111 11111111 00000000 00000000 00000000 00000000
11111111 00000000 11111111 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 ficher .bmp

TABLE – Format d'un bitmap

fichier	42 4D	Caractère B et M pour désigner un bitmap
	46 00 00 00	Taille du fichier 46 = 70 octets
	00 00 00 00	Réservé toujours à 0
	36 00 00 00	Position (offset) de l'image 36 = 54 octets
Entêtes image	28 00 00 00	Taille de l'entête du fichier 28 = 40 octets
	02 00 00 00	Largeur de l'image : 2 pixels
	02 00 00 00	Hauteur de l'image : 2 pixels
	01 00	nombre de plans utilisés
	18 00	nombre de bits par pixel 18 = 24 bits/pixel (3 octets)
	00 00 00 00	Méthode de compression (0 pas de compression)
	10 00 00 00	Taille de l'image 10 = 16 octets
	13 0b 00 00	Résolution horizontale 0b13 = 2835 pixel par mètre
	13 0b 00 00	Résolution verticale 0b13 = 2835 pixel par mètre
	00 00 00 00	Nombre de couleurs de la palette : 0 palette complète
	00 00 00 00	Nombre de couleurs importantes de la palette
Image	ff 00 00	B=255,V=0,R=0 : bleu
	ff ff ff	B=255, V=255, R=255 : blanc
	00 00	Pour que la ligne ait un nb d'octets multiple de 4
	00 00 ff	B=0, V=0, R=255 : rouge
	00 ff 00	B=0, V=255, R=0 : vert
	00 00	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 caclule 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→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.
  $5F_{hex} =_{dec} 5 \cdot 16 + F = 80 + 15 = 95$

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

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

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.

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.
  $4_{dec} =_{hex} 4, 9_{dec} =_{hex} 9, 12_{dec} =_{hex} C, 15_{dec} =_{hex} F$

Exercices $1 \to 1 \ldots 9 \to 9$ / $10 \to A$ / $11 \to B$ / $12 \to C$ / $13 \to D$ / $14 \to E$ / $15 \to F$

- Convertissez les nombres décimaux suivants en hexadécimal : 4, 9, 12, 15, 22, 24, 33, 102, 250, 4343 et 3853.
  $22_{dec} =_{hex} 1 \cdot 16 + 6 = 16$, $24_{dec} =_{hex} 1 \cdot 16 + 8 = 18$, $33_{dec} =_{hex} 2 \cdot 16 + 1 = 21$

Exercices $1 \to 1 \ldots 9 \to 9$ / $10 \to A$ / $11 \to B$ / $12 \to C$ / $13 \to D$ / $14 \to E$ / $15 \to F$

- 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 $1 \to 1 \ldots 9 \to 9$ / $10 \to A$ / $11 \to B$ / $12 \to C$ / $13 \to D$ / $14 \to E$ / $15 \to F$

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

- 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 $1\to 1 \ldots 9\to 9$ / $10\to A$ / $11\to B$ / $12\to C$ / $13\to D$ / $14\to E$ / $15\to F$

- 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→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.
  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

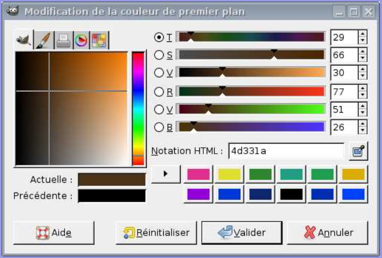
- En vraie couleur, donnez le code hexadécimal de la couleur comprenant 50% de rouge, 80% de vert et 3% de bleu.

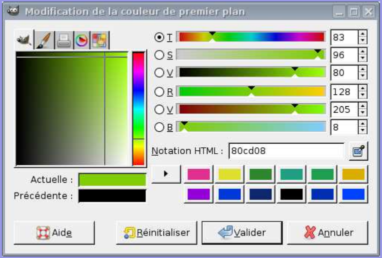
Exercices 1→1 ... 9→9 / 10→A / 11→B / 12→C / 13→D / 14→E / 15→F

- 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 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 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	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	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 $4d_H = 77_D = M_{ASCII}$
- 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 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 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
- 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 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 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
- 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	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		

- 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 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
- 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	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
- 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 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 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
```

- 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 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 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
```

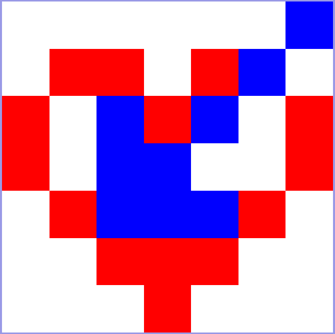
- BM car $42_H = 66_D = B_{ASCII}$ et $4d_H = 77_D = M_{ASCII}$
- $de_H = 222_D$ octets
- $36_H = 54_D$ octets
- $07_H = 7$ pixels
- $07_H = 7$ pixels
- $18_H = 24_D$ bits/pixel,soit 3 octets/pixel
- $a8_H = 168_D$ octets
- Un cœur percé.
- 222 octets, soit 0,217 ko

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	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		

- 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 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 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 00
```

Nombre total d'octet (deux chiffres hexadécimaux) :

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

Nombre de bits par pixel en vraie couleurs :

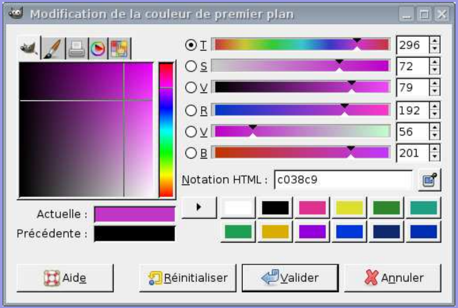
$$24_D = 18_H$$

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 suppémentaires 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)

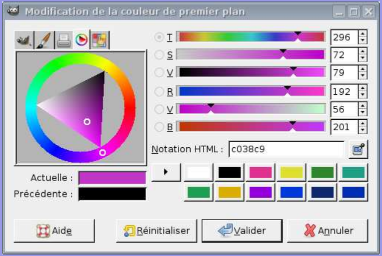
V 8 bits, 1 octet → 256 niveaux de vert

(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

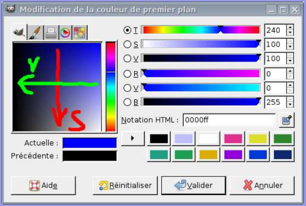


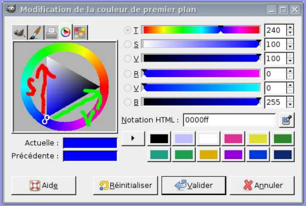
Teinte correspond à la couleur sur 360°. Le rouge se situe à 0°, le vert à 120° et le bleu à 240° selon un angle trigonométrique.

Saturation correspond à la quantité de couleur ″dans“ le blanc.

Valeur correspond, quant à elle, à la quantité de couleur ″dans“ le noir.

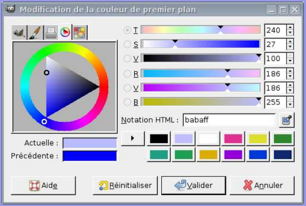
Exemples

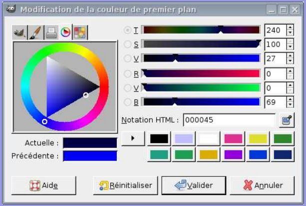




Bleu peu saturé

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 $T \in [0, 360]$.

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

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

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

Exercices

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

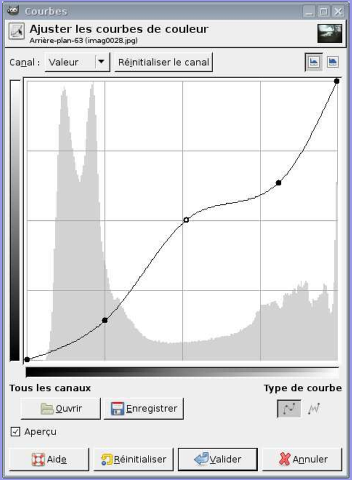
(34,56,239) ;
(22,240,200) ;
(255,231,176).

Exercices

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

(34,56,239) → (234,86,94) bleu très saturé et valorisé ;
(22,240,200) → (169,91,94) bleu-vert très saturé et vlorisé ;
(255,231,176) → (42,31,100) orange peu saturé et très valorisé.

L'outils courbe



Description :

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

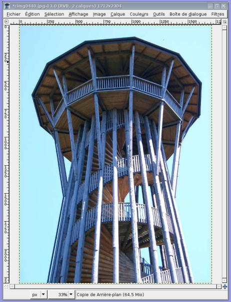
Ordonnée : Nombre de pixels de la valeur donnée.

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

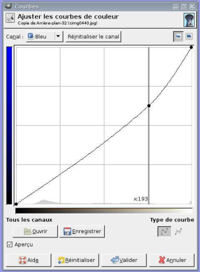
A gauche se trouve le *point noir* et à droite le *point blanc*.

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

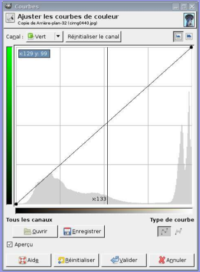
Original



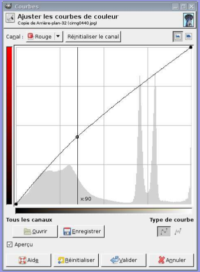
Répartition des tons bleus



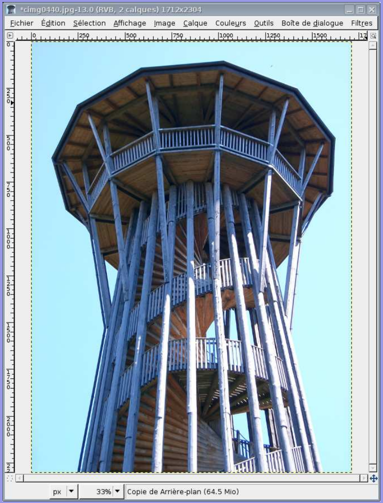
Répartition des tons verts

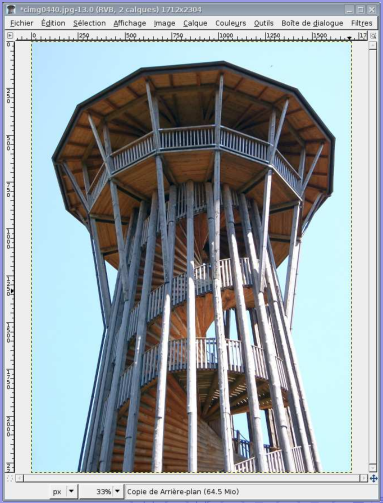


Répartition des tons rouges



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 occurences successives et sa valeur.

La chaine . . .

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 icones ou les documents au trait.

Compression de Huffmann : non destructive

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

Une chaine comprenant cinq lettres

ssionnansnni $\leftarrow$ oaisn

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

Un codage peu intelligent

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

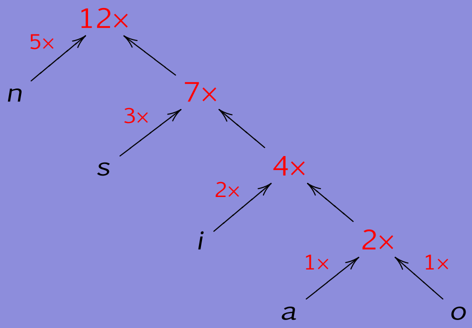
Le résultat est une chaine de $12 \cdot 3 = 36$ bits pour douze caractères.

La chaine encodée

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

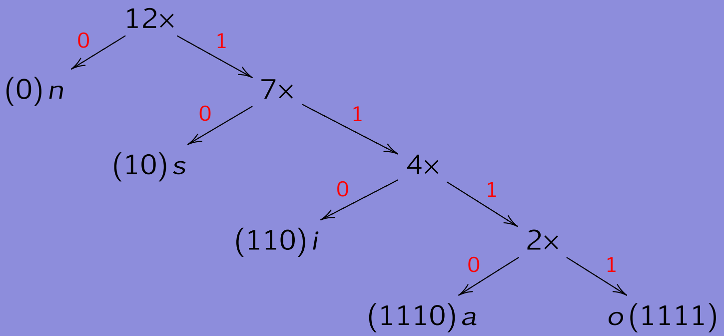
Méthode : éléments de plus basse fréquence 2 à 2.

Arbre de Huffman : construction



Méthode : 1 sur bit fort pour éléments basse fréquence.

Arbre de Huffman : codage



Codage de Huffmann

Selon l'arbre de Huffmann, 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

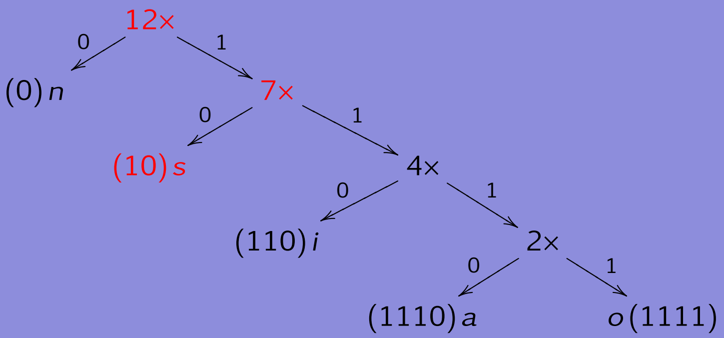
Codage de Huffmann

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 Huffmann.

Compression RLE pour le double F

```
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 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 00
```

16 x 14 - 2 = 222 octets

```
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 x 11 = 176 octets

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

Compression de Huffmann pour le double F

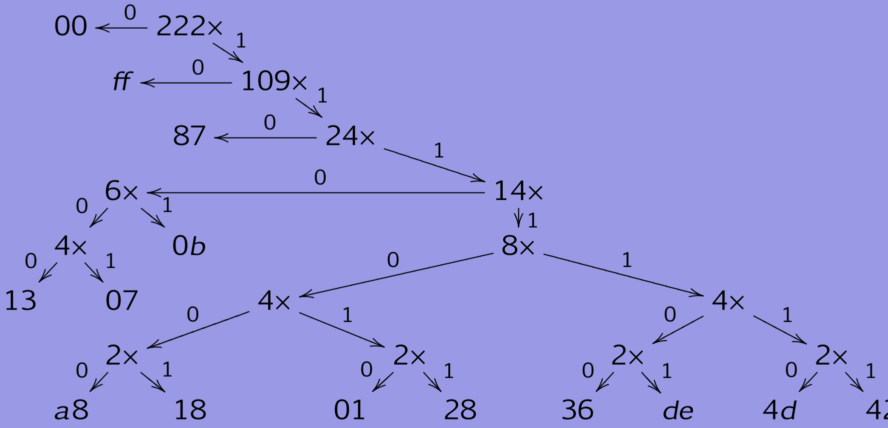
```
42 4d de 00 00 00 00 00 00 00 36 00 00 00 28 00
00 00 07 00 00 00 07 00 00 00 01 00 18 00 00 00
00 00 a8 00 00 00 13 0b 00 00 13 0b 00 00 00 00
00 00 00 00 00 00 00 ff 00 00 87 00 ff ff ff ff
ff ff ff ff ff ff ff ff ff ff ff 00 00 00 00 ff
00 00 87 00 ff ff ff ff ff ff ff ff ff ff ff ff
ff ff ff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

16 x 14 - 2 = 222 octets

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

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

Arbre de Huffmann



Codage de Huffmann

Arbre binaire => 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 (Adlus, 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 publique 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.

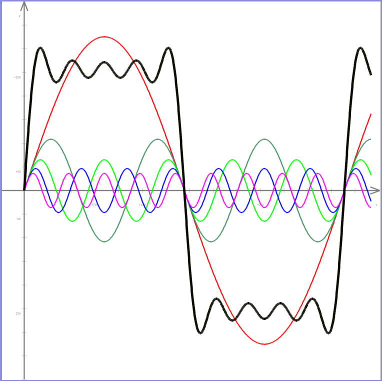
Principaux formats d'image : PDF

- **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.

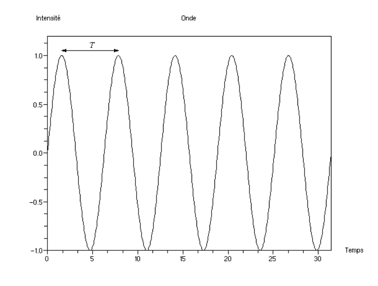
Table des matières

La physique du son

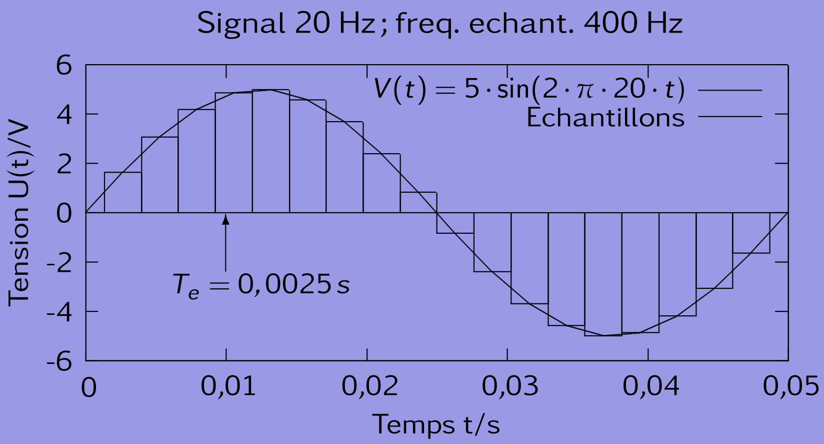
Décomposition de Fourrier[11]



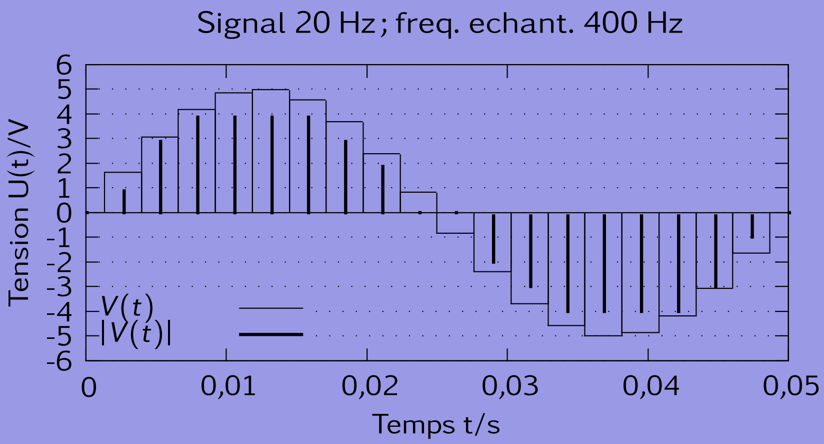
Onde sinusoidale[12]



Échantillonnage



Quantification



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	11
hex	0	1	3	4	4	4	4	3	2	0	0	?

Grandeurs importantes

Plusieurs grandeurs sont à considérer :

- durée du morçeau,
- fréquence maximale du signal
- fréquence d'échantillonage (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 : $f_e \geqslant 40'000\,Hz$ (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'échantillonage 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

Format : mp3

données organisées selon
le format standard mp3 :

entête, métadonnées (tag ID3 :
interprète, titre, nom de l'album, ...)
compression, ...

Extension : .mp3

La musique compressée

selon un codec
CODeur–DECodeur
MPEG–1/2 Audio Layer 3

Logiciel

Lecteur mp3 : amarok, xmms, ...

Encodeur – decodeur mp3
sous linux : lame

Principaux formats et codecs audio

WAV

Conteneur : mp3, pcm, wma ..., mais par référence non compressé : 10 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

Table des matières

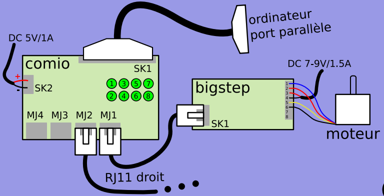
Exemple : “Un projet télescopique”

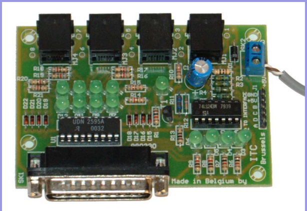
Objectif Réaliser le suivi d’une étoile guide par la motorisation du télescope du lycée.

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.

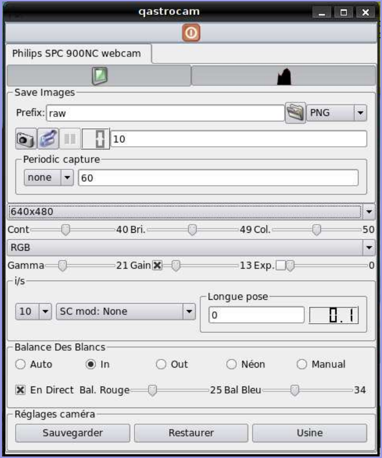
Rejetés Iris (fantastique logiciel gratuit de traitement d’images astro) car seuls les montures motorisées de type LX200 sont prévues. Code source fermé ⇒ accord et implication du développeur.





C++

$\Leftrightarrow$



Intégration du code

Code source ajouté à qastrocam :

Qastrocam

http://3demi.net/astro/qastrocam/doc/

```
#ifndef _QTelescopeLBC_hpp_
#define _QTelescopeLBC_hpp_
#include "QTelescope.hpp"
class PPort;
void *runEW(void *arg);
void *runSN(void *arg);
class QTelescopeLBC : public QTelescope {
    Q_OBJECT;
public:
    QTelescopeLBC(PPort * pport);
    void goE(float shift);
    void goW(float shift);
    void goS(float shift);
    void goN(float shift);
    void stopE();
    void stopW();
    void stopS();
    void stopN();
    void threads_start();
    void threads_stop();
    double setSpeed(double speed);
    bool setTracking(bool activated);
};
#endif
```

Limites du projet

Critiques importantes :

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

retour licences libres

Table des matières

Le contexte d'OpenOffice

Microprojet

Récupérer la page "informatique" de Wikipedia pour impression.

Le contexte d'OpenOffice

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.

OpenOffice, Xhtml, XML

Background de la mise en forme sous OO.

OO et XML

OO – XML

.odt $\leftrightarrows$ .zip

OpenOffice, Xhtml, XML

Background de la mise en forme sous OO.

OO et XML

OO – XML

.odt $\leftrightarrows$ .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.

Les transports

Pour les jeunes

Le skate

Planche à roulette

Le roller

Patins à roulette

le vélo

Deux roues

Pour les vieux

La voiture

Quatre roues

Le bateau

Une coque

Traduction en XML

Retour OpenOffice

```
<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>
```

Table des matières

Premier semestre : deux évaluations écrites

Projet TEXTE

- Rendre avant Noël,
- Seul ou a deux (même note),
- Utilisation de Scribus ou LaTeX ou Code HTML,
- Une seule feuille (recto-verso ; deux pages HTML) réalisée essent. pendant le cours,
- Une page expliquant les problèmes rencontrés,
- Une licence expliquée,
- Tout compte.

Travail écrit Sur le cours depuis le début de l'année, le premier jeudi de la rentrée de Noël.

Second semestre : deux évaluations (orale ; écrite)

Un projet à réaliser sur un semestre. Trois personnes par projet. Validation du projet le premier jeudi du second semestre par l'enseignant.

Projet présentation Rapport oral intermédiaire présentant le travail accomplis, les problèmes rencontrés et ce qu'il reste à faire.

Projet bilan Restitution du projet.

Retour cours

Table des matières

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, LaTeXn'est pas un language XML!

Le préambule

Voici un exemple de préambule commenté
(en LaTeXles 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 francaise
\usepackage{verbatim}             % Pour mettre des commentaires etendus
\usepackage{graphicx}             % Pour mettre des images
\graphicspath{{./Images/}}        % Pour eviter de mettre le chemin des images
\usepackage{fancyhdr}             % Pour des entetes
\fancyhead[RE,LO]{Lycee Blaise Cendrars\\La Chaux-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 elargir un peu l'entete
```

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 LaTeXtient 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
```

Le résultat :

- Truc
- Chose

Table des matières

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.

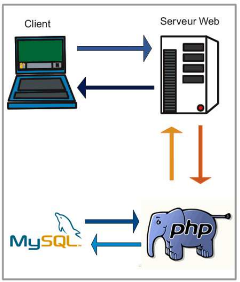


Table des matières

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 $\rightarrow$ quelques interactions aspects visuels - contenu. Problème : publicités.
2. CMS à installer. Le contôle du code $\rightarrow$ 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 $\rightarrow$ 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

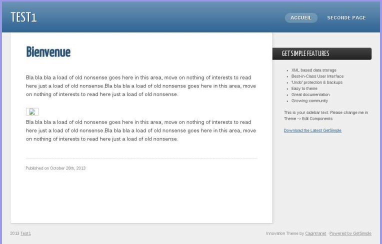
Installation simple par dépôt ftp (sous filezilla par exemple).

FTP : File Transfert Protocol

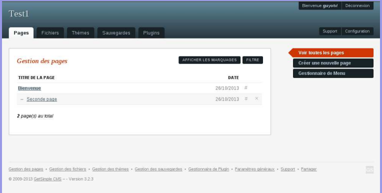
Il s'agit d'un ensemble de règles de transfert d'information sur le réseau par copie de fichiers. On peut parler des outils nécessaires pour gérer des opérations de navigateur de fichiers sur des machines distantes.

Résultat

Une page d'accueil



Une zone d'administration



Structure template

Table des matières

Versions

Il existe plusieurs versions de HTML qui ne sont pas compatibles avec tous les navigateurs.

En 2014, la norme *HTML5* devrait être officialisée. C'est donc sur celle-ci qu'on va se baser. Elle est compatible avec tous les navigateurs récents dans la pluspart des cas.

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>

Balisage

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 />,
 (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 :

 Wikipedia

Ligne horizontale :

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

Image :

Image - lien :

Google

Structure générale

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

<!DOCTYPE html>	–> la déclaration du type HTML5 du document
<html>	–> balise d'ouverture du document
<head>	–> balise d'ouverture de l'entête
...	...
</head>	–> balise de fermeture de l'entête
<body>	–> balise d'ouverture du corps du document
...	...
</body>	–> balise de fermeture du corps du document
</html>	–> balise de fermeture du document

Structure générale

La structure générale de l'entête est la suivante :

```
<head>
  <meta charset="UTF-8" />
  <meta name="description"
        content="Description␣
            page" />
  <meta name="author"
        content="mon␣nom" />
  <title>Titre de la fenetre</
      title>
  <link rel="stylesheet"
        href="style.css" />
  <style> ... </style>
  <script> ... </script>
</head>
```

–> balise d'ouverture de l'entête
–> Codage des caractères utilisés
–> balise de description de la page
...
–> balise de déclaration de l'auteur
...
–> balise de titre de la fenêtre
–> balise de feuille de style externe
...
–> balise de feuille de style interne
–> balise de script
–> balise de fermeture de l'entête

Structure générale

La structure générale du corps est la suivante :

```
<body>                                  –> balise d'ouverture du corps
  <header> ... </header>                –> balise de l'entête (logo, titre, ...)
  <nav> ... </nav>                      –> balise de la barre de navigation
  <aside> ... </aside>                  –> balise de colonne latérale droite
  <article> ... </article>              –> balises des articles
  ...                                   –> ...
  <footer> ... </footer>                –> balise du pied de page (copyright, ...)
</body>                                 –> balise de fermeture du corps
```

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 :

$\boxed{\text{sélecteur \{propriété :valeur\}}}$

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 très 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="laclass">La </h1>
<em class="laclass">deux chevaux,
<p class="laclass">sa puissance </p>
<p>c'est la lenteur.</p>

.laclass {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>
deux chevaux,
<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.

Table des matières

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 tablau 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
    }
    spriteImage.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.backgroundPosition =
         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 publique. 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.

[1] Wikipedia : http://commons.wikimedia.org/wiki/File:Phoenix_on_the_Red_Planet.jpg
[2] http://fr.wikipedia.org/wiki/Fichier:Architecture_micro-noyau.png
[3] http://upload.wikimedia.org/wikipedia/fr/f/f9/Architecture_noyau-monolithique.png
[4] Wikipedia : http://fr.wikipedia.org/wiki/Logiciels_libres
[5] Wikipedia : https://upload.wikimedia.org/wikipedia/commons/thumb/8/8b/Raspberry_Pi_4%2C_2_GB_RAM_version.
[6] Wikipedia : http://commons.wikimedia.org/wiki/File:Notes.svg
[7] Voir wikipedia : http://commons.wikimedia.org/wiki/File:CCD_in_camera.jpg
[8] Voir wikipedia : http://commons.wikimedia.org/wiki/File:Pixel.jpg
[9] Voir wikipedia : http://fr.wikipedia.org/wiki/Fichier:Resolution_test.jpg
[10] Voir wikipedia : http://commons.wikimedia.org/wiki/File:BayerPatternFiltration.png
[11] Voir Wikicommon : http://commons.wikimedia.org/wiki/File:Fourier_d%27un_carr%C3%A9.svg
[12] Voir Wikicommon : http://commons.wikimedia.org/wiki/File:Onde_endroit_fixe.png