

Web

Bruno BEAUFILS

2021/2022

1. Introduction

À propos du cours

À propos d'informatique

Références

1. Introduction

À propos du cours

À propos d'informatique

Références

- ▶ Maître de conférences en informatique
- ▶ Université de Lille
 - ▶ IUT / Département informatique
 - ▶ Systèmes d'exploitation
 - ▶ Réseaux
 - ▶ CRIStAL (*Centre de Recherche en Informatique, Signal et Automatique de Lille*)
 - ▶ Systèmes Multi-Agents (Théorie des jeux, Finance computationnelle)
 - ▶ Intelligence artificielle distribuée / Vie artificielle
 - ▶ Réseaux de co-auteurs
 - ▶ Médiation scientifique
- ▶ Membre du CA de la [Société Informatique de France](#)
- ▶ Contacts
 - ▶ email : bruno.beaufils@univ-lille.fr
 - ▶ matrix : [@beaufils:matrix.org](matrix://bruno.beaufils.org)
 - ▶ web : <https://beaufils.u-lille.fr>
 - ▶ twitter : [@brunobeaufils](https://twitter.com/brunobeaufils)
 - ▶ mastodon : [@beaufils@mastodon.social](https://mastodon.social/@beaufils)

- ▶ Horaire : 14h à 17h15 en DEM.027
- ▶ Séances :
 1. 29/10
 2. 12/11 (9h à 12h en DEM.309)
 3. 19/11
 4. 26/11
 5. 03/12
 6. 10/12
 7. 17/12 examen
- ▶ Ressources

<https://m2.ape-cee.fr>

Prérequis

- ▶ Avoir déjà manipulé un système d'exploitation sur ordinateur de bureau (Windows, Mac OS X, GNU/Linux)
- ▶ Connaître la notion de fichiers

Outils

- ▶ Quasiment chaque séance sera accompagné de travaux pratiques.
- ▶ Les outils suivants seront utilisés :
 - ▶ éditeur de texte
 - ▶ navigateur web
 - ▶ outils en ligne de commandes

Modalités d'évaluation (année « normale »)

- ▶ **1 note de devoir** (*coefficient 0,5*)
travaux rendus (débuté en séance, fini chez vous)
- ▶ **1 examen écrit** (*coefficient 0,5*)
évaluation faite lors de la dernière séance

1. Comprendre le fonctionnement de l'Internet
 - ▶ techniquement
 - ▶ administrativement
2. Comprendre le fonctionnement de quelques services de l'Internet : *web* et *DNS*
 - ▶ techniquement
 - ▶ pratiquement
3. Savoir manipuler les services *web*
 - ▶ produire
 - ▶ contenu HTML
 - ▶ mise en forme CSS (Javascript)
 - ▶ consommer
 - ▶ récupérer des données via les sélecteurs CSS
4. Aborder l'aspect *politique* de l'Internet
 - ▶ comprendre les effets de bord du fonctionnement du web et du DNS
 - ▶ industrie des données (personnelles)
 - ▶ implication sur la vie privée
 - ▶ comprendre le mécanisme de gestion de l'Internet

La partie *manipulation du web* sera la partie la plus importante en volume.

1. Introduction

À propos du cours

À propos d'informatique

Références

Définition

- ▶ en Français
- ▶ en Anglais

Traitement automatique de l'information

Computer science

Science des calculateurs

Outils pour

- ▶ représenter l'information
 - ▶ mathématiques / philosophie
- ▶ manipuler cette représentation
 - ▶ machine électrique (électronique)

Science à mi-chemin entre les concepts (*mathématique*) et le réel (*physique*)

Constitutants de l'informatique

En informatique il y a :

1. Machine
2. Algorithme
3. Langage
4. Information
5. *Droits*

Machine = Ordinateur

Une simple machine à calculer

- ▶ un processeur
- ▶ une horloge
- ▶ de la mémoire
 - ▶ principale (*RAM*)
 - ▶ secondaire (*disque*)
- ▶ des périphériques de communication

calcul
temporisation
stockage
non-durable
durable
interaction

Exemples :

- ▶ super-calculateur
- ▶ serveur
- ▶ micro-ordinateur portable
- ▶ téléphone (*smart*-phone ou plutôt **ordi**-phone)
- ▶ objets connectés

Codage de l'information

Toute information peut se représenter par un nombre

- ▶ Utilisation fréquente de **table** pour énumérer les informations
 - ▶ Liste indexée par un nombre entier
 - ▶ Exemple : UCS = Unicode Character Set (jeu de caractères universel)
- ▶ Utilisation d'algorithmes pour la **représentation** de l'index avec des contraintes
 - ▶ Contraintes fréquentes
 - ▶ taille des nombres utilisables
 - ▶ Exemple : UTF-8 = Unicode Transformation Format

Numérique \neq Digital

- ▶ représentation en base 2 {0,1}
 - ▶ chiffres binaires = **bits** (*binary digits*)
- ▶ représentation en base 16 {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}
 - ▶ chiffres hexadécimaux
- ▶ nombre limités à 8 bits **octet**
 - ▶ *byte* en anglais
 - ▶ nombre compris entre 0 et 255
 - ▶ représentable avec 2 chiffres hexadécimaux
- ▶ système positionnel : poids d'un chiffre de nombre
 - ▶ **fort** (à gauche) et **faible** (à droite)

Exercice : conversions entre bases

- ▶ Convertissez les nombres suivants de la base 10 à la base 2
 - ▶ 42
 - ▶ 19
- ▶ Convertissez les nombres suivants de la base 2 à la base 10
 - ▶ 10011100
 - ▶ 00001110
- ▶ Écrivez tous les nombres précédents en base 16

Aspect important de l'informatique : *science du calcul*

- ▶ Description des instructions pour effectuer un *calcul*
- ▶ Expression dans un langage spécifique

Concepts

1. Variable
2. Séquence
3. Alternative
4. Boucles
 - ▶ répétition basée sur un test
 - ▶ répétition basée sur une liste de valeur
5. Fonctions

Langages de programmation

- ▶ syntaxe
- ▶ sémantique (impératif, orientée objet, fonctionnel, déclaratif, logique, etc.)
- ▶ bibliothèques de fonctions et interface d'utilisation (*API*)
- ▶ exemple : Python, Javascript, C, R, etc.

Aspect important de l'informatique : *science du calcul*

- ▶ Description des instructions pour effectuer un *calcul*
- ▶ Expression dans un langage spécifique

Concepts

1. Variable
2. Séquence
3. Alternative
4. Boucles
 - ▶ répétition basée sur un test
 - ▶ répétition basée sur une liste de valeur
5. Fonctions

Langages de programmation

- ▶ syntaxe
- ▶ sémantique (impératif, orientée objet, fonctionnel, déclaratif, logique, etc.)
- ▶ bibliothèques de fonctions et interface d'utilisation (*API*)
- ▶ exemple : Python, Javascript, C, R, etc.

Aspect important de l'informatique : *science du calcul*

- ▶ Description des instructions pour effectuer un *calcul*
- ▶ Expression dans un langage spécifique

Concepts

1. Variable
2. Séquence
3. Alternative
4. Boucles
 - ▶ répétition basée sur un test
 - ▶ répétition basée sur une liste de valeur
5. Fonctions

Langages de programmation

- ▶ syntaxe
- ▶ sémantique (impératif, orientée objet, fonctionnel, déclaratif, logique, etc.)
- ▶ bibliothèques de fonctions et interface d'utilisation (*API*)
- ▶ exemple : Python, Javascript, C, R, etc.

Structuration système

- ▶ Suite **non structurée** d'octets (de nombre)
- ▶ 2 types importants
 1. Dossier (*répertoire*)
 2. Document (*fichier*)
- ▶ Notion de hiérarchie
 - ▶ sous Unix
 - ▶ une seule (/)
 - ▶ sous Windows
 - ▶ une par volume (C: \, D: \, etc.)

(contient des documents)
(contient des données)

Structuration applicative

format

- ▶ **convention** d'agencement des données en interne
 - ▶ exemples :
 - ▶ texte (.txt),
 - ▶ bureautique (ODF, OOXML),
 - ▶ image (JPG, PNG),
 - ▶ archive (ZIP)
 - ▶ outils adaptés à chaque convention (éditeur de texte, formatteur, visualisateur)
 - ▶ notion de **type** de contenu
- ▶ indépendance du nom (de l'extension) et du type de contenu
- ▶ *souvent* utilisation du nom pour
 - ▶ déterminer la convention
 - ▶ déterminer l'application à utiliser pour manipuler le contenu
- ▶ **le système cache les extensions**
- ▶ éditeur de textes
 - ▶ Multi systèmes (Windows/MacOS/Linux) : bluefish, Atom, Brackets, Emacs, Visual Studio Code
 - ▶ Windows : Bloc notes, Notepad++

Structuration applicative

format

- ▶ **convention** d'agencement des données en interne
 - ▶ exemples :
 - ▶ texte (.txt),
 - ▶ bureautique (ODF, OOXML),
 - ▶ image (JPG, PNG),
 - ▶ archive (ZIP)
 - ▶ outils adaptés à chaque convention (éditeur de texte, formatteur, visualisateur)
 - ▶ notion de **type** de contenu
- ▶ indépendance du nom (de l'extension) et du type de contenu
- ▶ *souvent* utilisation du nom pour
 - ▶ déterminer la convention
 - ▶ déterminer l'application à utiliser pour manipuler le contenu
- ▶ **le système cache les extensions**
- ▶ éditeur de textes
 - ▶ Multi systèmes (Windows/MacOS/Linux) : [bluefish](#), [Atom](#), [Brackets](#), [Emacs](#), [Visual Studio Code](#)
 - ▶ Windows : Bloc notes, [Notepad++](#)

Structuration applicative

format

- ▶ **convention** d'agencement des données en interne
 - ▶ exemples :
 - ▶ texte (.txt),
 - ▶ bureautique (ODF, OOXML),
 - ▶ image (JPG, PNG),
 - ▶ archive (ZIP)
 - ▶ outils adaptés à chaque convention (éditeur de texte, formatteur, visualisateur)
 - ▶ notion de **type** de contenu
- ▶ indépendance du nom (de l'extension) et du type de contenu
- ▶ *souvent* utilisation du nom pour
 - ▶ déterminer la convention
 - ▶ déterminer l'application à utiliser pour manipuler le contenu
- ▶ **le système cache les extensions**
- ▶ éditeur de textes
 - ▶ Multi systèmes (Windows/MacOS/Linux) : [bluefish](#), [Atom](#), [Brackets](#), [Emacs](#), [Visual Studio Code](#)
 - ▶ Windows : [Bloc notes](#), [Notepad++](#)

Structuration applicative

format

- ▶ **convention** d'agencement des données en interne
 - ▶ exemples :
 - ▶ texte (.txt),
 - ▶ bureautique (ODF, OOXML),
 - ▶ image (JPG, PNG),
 - ▶ archive (ZIP)
 - ▶ outils adaptés à chaque convention (éditeur de texte, formatteur, visualisateur)
 - ▶ notion de **type** de contenu
- ▶ indépendance du nom (de l'extension) et du type de contenu
- ▶ *souvent* utilisation du nom pour
 - ▶ déterminer la convention
 - ▶ déterminer l'application à utiliser pour manipuler le contenu
- ▶ **le système cache les extensions**
- ▶ éditeur de textes
 - ▶ Multi systèmes (Windows/MacOS/Linux) : **bluefish**, **Atom**, **Brackets**, **Emacs**, **Visual Studio Code**
 - ▶ Windows : Bloc notes, **Notepad++**

Exercice : manipulation de fichiers

Créez une hiérarchie sur votre Bureau d'après votre identifiant

```
beaufils/  
+-- divers  
+-- email  
+-- experiences  
+-- formation  
+-- identites  
+-- naissance/  
| +-- date  
| `-- lieu  
+-- nom  
+-- photo.url  
`-- prenom
```

- ▶ ne mettez **pas de suffixe** (extension) à vos noms de fichiers
 - ▶ **respectez la casse** (minuscule/majuscule) à vos noms de fichiers
 - ▶ naissance est un **dossier** les autres des fichiers textes
 - ▶ email ne contient que votre adresse email
 - ▶ photo.url contient un lien vers une de vos photos
 - ▶ identites contient vos identités sur les réseaux sociaux (Twitter, Facebook, etc.)
- ▶ chaque ligne est construite sur l'exemple :
twitter:brunobeaufils

Envoyez-moi votre travail

- ▶ dans une archive au format .zip contenant votre dossier
- ▶ archive nommée d'après votre identifiant
- ▶ adresse : **bruno@ape-cee.fr**
- ▶ sujet : **[M2APE] Prénom NOM**
- ▶ le **strict** respect de ces contraintes est important

Définition

C'est une suite d'instructions pour traiter de l'information sur un processeur :

- ▶ faire des calculs
- ▶ stocker les résultats
- ▶ utiliser ces résultats

Un logiciel ne vit pas seul

- ▶ interaction avec des humains
- ▶ communication avec d'autres logiciels

Élaboration d'un logiciel

- ▶ **auteur** *éditeur*
- ▶ **code source** *langage de programmation*
- ▶ **code exécutable** *binaire*

Un **éditeur** est **propriétaire** de ce qu'il produit

- ▶ protégé par le droit d'auteur (ou le **copyright** ©)
- ▶ libre de les (droits) utiliser comme bon lui semble

Un **utilisateur** est **tributaire** de la volonté de l'éditeur dans le cas où

- ▶ il n'a pas les compétences d'édition
- ▶ il a besoin du service rendu par le logiciel

Éditeur et utilisateur sont liés par un *contrat* : **une licence**

Caractéristiques d'une licence

Un logiciel a une licence

- ▶ **libre** si elle permet de :
 1. l'**utiliser sans restrictions**
 2. l'**étudier** et de le **modifier**
 3. le **redistribuer**
 4. le **distribuer des versions modifiées**
- ▶ **privatrice** si elle restreint une de ces 4 libertés
 - ▶ souvent nommé CLUF (*Contrat de Licence Utilisateur Final*)
 - ▶ précise souvent :
 - ▶ l'utilisation prévue
 - ▶ l'interdiction de la rétro-ingénierie
 - ▶ l'interdiction de la copie

On essaiera de n'utiliser que des logiciels libres

Caractéristiques d'une licence

Un logiciel a une licence

- ▶ **libre** si elle permet de :
 1. l'**utiliser sans restrictions**
 2. l'**étudier** et de le **modifier**
 3. le **redistribuer**
 4. le **distribuer des versions modifiées**
- ▶ **privatrice** si elle restreint une de ces 4 libertés
 - ▶ souvent nommé CLUF (*Contrat de Licence Utilisateur Final*)
 - ▶ précise souvent :
 - ▶ l'utilisation prévue
 - ▶ l'interdiction de la rétro-ingénierie
 - ▶ l'interdiction de la copie

On essaiera de n'utiliser que des logiciels libres

Caractéristiques d'une licence

Un logiciel a une licence

- ▶ **libre** si elle permet de :
 1. l'**utiliser sans restrictions**
 2. l'**étudier** et de le **modifier**
 3. le **redistribuer**
 4. le **distribuer des versions modifiées**
- ▶ **privatrice** si elle restreint une de ces 4 libertés
 - ▶ souvent nommé CLUF (*Contrat de Licence Utilisateur Final*)
 - ▶ précise souvent :
 - ▶ l'utilisation prévue
 - ▶ l'interdiction de la rétro-ingénierie
 - ▶ l'interdiction de la copie

On essaiera de n'utiliser que des logiciels libres

Extension du domaine de la lutte : Creative Commons

Principes posés par [Lawrence LESSIG](#) en 2002 :

- ▶ Appliquer les principes du logiciel libre à toutes les types œuvres de l'esprit
 - ▶ notamment en ligne
- ▶ S'appuyer sur la volonté de l'auteur pour faciliter la réutilisation créative, le remix, etc
- ▶ Passer d'une approche « **Tous droits réservés** » à « **certains droits réservés** »

Un système de licences modulables en 4 pôles

- ▶ Attribution de Paternité CC 
- ▶ Pas d'utilisation commerciale NC  
- ▶ Pas de Modification ND 
- ▶ Partage dans les Mêmes Conditions SA 

Au final 6+1 licences différentes, avec des signalétiques adaptées



Extension du domaine de la lutte : Creative Commons

Principes posés par [Lawrence LESSIG](#) en 2002 :

- ▶ Appliquer les principes du logiciel libre à toutes les types œuvres de l'esprit
 - ▶ notamment en ligne
- ▶ S'appuyer sur la volonté de l'auteur pour faciliter la réutilisation créative, le remix, etc
- ▶ Passer d'une approche « **Tous droits réservés** » à « **certains droits réservés** »

Un système de licences modulables en 4 **pôles**

- ▶ **Attribution de Paternité** CC 
- ▶ **Pas d'utilisation commerciale** NC  
- ▶ **Pas de Modification** ND 
- ▶ **Partage dans les Mêmes Conditions** SA 

Au final 6+1 licences différentes, avec des signalétiques adaptées



Extension du domaine de la lutte : Creative Commons






















Principes posés par [Lawrence LESSIG](#) en 2002 :

- ▶ Appliquer les principes du logiciel libre à toutes les types œuvres de l'esprit
 - ▶ notamment en ligne
- ▶ S'appuyer sur la volonté de l'auteur pour faciliter la réutilisation créative, le remix, etc
- ▶ Passer d'une approche « **Tous droits réservés** » à « **certains droits réservés** »

Un système de licences modulables en 4 **pôles**

- ▶ **Attribution de Paternité** CC 
- ▶ **Pas d'utilisation commerciale** NC  
- ▶ **Pas de Modification** ND 
- ▶ **Partage dans les Mêmes Conditions** SA 

Au final 6+1 licences différentes, avec des signalétiques adaptées

CC-0	CC-BY	CC-BY-SA	CC-BY-ND	CC-BY-NC	CC-BY-NC-SA	CC-BY-NC-ND
 	 	  	  	  	   	   

- ▶ Accès aux services de l'Internet
 - ▶ web : **Firefox**
 - ▶ email : **Thunderbird**
 - ▶ distribution de fichiers : **TransmissionBT**
- ▶ Manipulation multimédia
 - ▶ création : **Gimp**, **Krita**, **Inkscape**, **Blender**
 - ▶ visualisation : **VLC**
- ▶ Gestion bureautique : **LibreOffice**
- ▶ Outils scientifiques
 - ▶ calculs : **R**, **Scilab**
 - ▶ traitement de textes : **T_EX**
- ▶ Jeux
 - ▶ **Minetest**
 - ▶ **FlightGear Flight Simulator**
 - ▶ **GCompris**

Une liste de logiciels libres est disponible sur [Framalibre](#)

1. Introduction

À propos du cours

À propos d'informatique

Références

Pour aller plus loin

- ▶ Apprentissage de l'algorithmique et de la programmation
 - ▶ <http://www.france-ioi.org/algo/index.php>
- ▶ Les licences logiciels (droit, informatique, et libertés)
 - ▶ <https://bruno.boulgour.com/talks/2019-06-20-diu-eil-libre/>