

- ▶ Organisation du S6
- ▶ Une L, et après ?

L3 informatique

18 janvier 2024

# **Le semestre 6**

# Semestre 6 - L3 info

Matière	ref	ECTS
Prog. réseaux	PR6	6
Grammaires et analyse syntaxique	GA6	6
Base de données	BD6	6
Anglais		3
<b>Bloc options</b>		<b>9</b>

# Options du S6

Matière			
Intro. Genie logiciel	3	24 h CTP	
Lang. de script	3	24 h CTP	
Machines virtuelles	3	24 h CTP	
Prog. efficace	3	24 h CTP	
Prog. web	3	24 h CTP	
Sécurité info	3	24 h CTP	
Cryptographie	3	24 h CTP	
Probabilités	6	24 h CM + 24h TD	22/01 !
Bioinformatique	3	24 h CTP	22/01 !
Linguistique	3	24 h CTP	

**(9 ects à choisir...)**

# Le choix des options

# Procédure

Vous aurez à choisir (via le serveur pédagogique):

- **votre groupe de TD/TP** pour les 3 UEs obligatoires (prog réseaux, BD, Grammaire et analyse) et donc un **EDT**,
- **vos options** (des UE et des créneaux horaires).

Ces deux choix, avec votre cours d'anglais, définira complètement votre emploi du temps du deuxième semestre.

Le serveur ouvrira aujourd'hui à 14h.

**NB:** pour les étudiant-e-s ayant peu d'UE à repasser ou les étudiants AJAC, la procédure d'inscription se fera directement à la scolarité à partir de lundi 22 janvier.

# Procédure

Inscription via le serveur pédagogique jusqu'à lundi 12h. ou à la scolarité à partir de lundi 22 janvier.

Vous pourrez alors choisir le groupe et les options qui vous conviennent le mieux (en fonction de la place !).

**Vous n'aurez pas forcément votre groupe de TD préféré ni vos options préférées. Et ce ne sera pas un drame !**

## **Pour les redoublants:**

- pas d'obligation à refaire les même options.
- il est possible que l'on vous refuse certaines options très demandées si vous les avez déjà suivies l'année dernière.

# Les options



# Probabilités

N. Cunéo / C. Labbé

CM+TD : 6 ects

Premier CM: semaine du 22 janvier !!



# Contenu du cours

L'informatique pour les langues naturelles / humaines :

1. Introduction
  - ▶ Généralités sur la langue naturelle
  - ▶ Langues, langages, automates
2. Reconnaissance de l'écriture et de la parole
  - ▶ Acquisition (numérisation) de données textuelles
  - ▶ Reconnaissance (parole ASR / écrit OCR)
3. Extraction d'informations et indexation de documents
  - ▶ Représentation de documents
  - ▶ Moteurs de recherche (basés sur le contenu)
4. Analyse et représentation de la langue
  - ▶ Compréhension de la langue humaine (morphologie, syntaxe)
5. Traduction automatique et assistée
  - ▶ Mécanismes de la traduction automatique
6. Agents dialogiques
  - ▶ Principe des chatbots
  - ▶ Interactions utilisateur en langue naturelle

# Prérequis et perspectives

- ▶ Prérequis :
  - ▶ Intérêt pour les langues humaines !
  - ▶ Traitements orientés données (volumineuses)
  - ▶ Niveau minimal en programmation (exercices)
- ▶ Quelques perspectives :
  - ▶ Fondements des logiciels de TAL (NLP)
  - ▶ Mécanismes d'apprentissage automatique
  - ▶ Sciences des données
  - ▶ Analyse et compréhension de la langue naturelle
  - ▶ Multilinguisme et langues peu dotées
  - ▶ Poursuite en master TAL (Diderot, Inalco, etc.)

# Cryptographie

R. Amadio

## Sécurité informatique et cryptographie

Sécurité informatique  $\supset$  Crypto-logie  $\left\{ \begin{array}{l} \text{Crypto-graphie} \\ \text{Crypto-analyse} \end{array} \right.$

On utilisera **cryptographie** pour les deux activités ; il s'agit de la partie de la sécurité informatique :

- qu'on 'contourne',
- dans laquelle on fait des 'preuves' et on gagne des prix Turing : Blum (1995), Yao (2000), Rivest-Shamir-Adleman (2002), Micali-Goldwasser (2012), Diffie-Hellman (2015).

# Cryptographie: qu'est ce que c'est?

- Une (petite) partie de sécurité informatiques.
- Deux propriétés au centre du cours :
  - **confidentialité** (Roberto veut transmettre à Eugène le sujet de l'examen, sans que les étudiants puissent l'avoir). Principe: chiffrer-déchiffrer
  - **Intégrité** (Eugène veut être sûr que le message vient de Roberto) Principe: signer-vérifier
- On les définira et on verra comment les assurer
- Approche classique (clé secrète) et approche moderne (clé publique)
- On prouvera que nos solutions marchent (sous certaines hypothèses)

# Crypto - organisation

- 2 groupes indépendants (A et B) , 2 enseignants (Roberto Amadio, Eugene Asarin), 95% même programme, même examen.
- Seulement un C/TD par semaine, mais travail systématique hors cours (« TD » et « TP » ) requis.
- DM à rendre contient des TPs sur la totalité du programme. À programmer en équipe dans votre langage préféré (Python, Java...).
- Examen fortement inspiré par les TD/TP
- Moodle avec supports détaillés (regardez-le, il n'est pas verrouillé)

# Pré-requis

- Notions basiques de *(mais on rappelle tout)*:
  - probabilités (discrètes)
  - arithmétique (modulaire, nombres premiers etc)
  - algèbre (groupes, anneaux)
  - algorithmique/complexité d'algo
- Pas d'allergie aux raisonnements mathématiques
- Certaine maîtrise de programmation (*partie TP 100% autonome*).

*Suivre la Sécurité en même temps conseillé mais pas indispensable*



# Cryptographie

R. Amadio

## Objectifs du cours

1. Définir les propriétés de **confidentialité** et d'**intégrité**.
2. Expliquer comment assurer ces propriétés à partir de certaines **hypothèses** comme “l'exposant modulaire est une fonction à sens unique”.

**NB** On a juste le temps de traiter les fonctions cryptographiques de base... mais on verra des exemples de protocoles.

# Cryptographie: qu'est ce que c'est?

- Une (petite) partie de sécurité informatiques.
- Deux propriétés au centre du cours :
  - **confidentialité** (Roberto veut transmettre à Eugène le sujet de l'examen, sans que les étudiants puissent l'avoir). Principe: chiffrer-déchiffrer
  - **Intégrité** (Eugène veut être sûr que le message vient de Roberto) Principe: signer-vérifier
- On les définira et on verra comment les assurer
- Approche classique (clé secrète) et approche moderne (clé publique)
- On prouvera que nos solutions marchent (sous certaines hypothèses)

# Crypto - organisation

- 2 groupes indépendants (A et B) , 2 enseignants (Roberto Amadio, Eugene Asarin), 95% même programme, même examen.
- Seulement un C/TD par semaine, mais travail systématique hors cours (« TD » et « TP » ) requis.
- DM à rendre contient des TPs sur la totalité du programme. À programmer en équipe dans votre langage préféré (Python, Java...).
- Examen fortement inspiré par les TD/TP
- Moodle avec supports détaillés (regardez-le, il n'est pas verrouillé)

# Pré-requis

- Notions basiques de *(mais on rappelle tout)*:
  - probabilités (discrètes)
  - arithmétique (modulaire, nombres premiers etc)
  - algèbre (groupes, anneaux)
  - algorithmique/complexité d'algo
- Pas d'allergie aux raisonnements mathématiques
- Certaine maîtrise de programmation (*partie TP 100% autonome*).

*Suivre la Sécurité en même temps conseillé mais pas indispensable*

# Cryptographie

R. Amadio

## Organisation

- 12 séances.
- **Cours :**
  - Diapos + poly sur la page du cours (à regarder avant, à étudier après).
  - Séance = cours+proposition exercices+discussion  
exercices proposés.
- **MCCC : Contrôle Continu Intégral**
  - Rendre la solution de certains exercices proposés (en équipe).
  - Évaluer la solution rendue par une autre équipe (en équipe).
  - Examen écrit (individuel) qui porte en priorité sur les exercices proposés.

# Cryptographie: qu'est ce que c'est?

- Une (petite) partie de sécurité informatiques.
- Deux propriétés au centre du cours :
  - **confidentialité** (Roberto veut transmettre à Eugène le sujet de l'examen, sans que les étudiants puissent l'avoir). Principe: chiffrer-déchiffrer
  - **Intégrité** (Eugène veut être sûr que le message vient de Roberto) Principe: signer-vérifier
- On les définira et on verra comment les assurer
- Approche classique (clé secrète) et approche moderne (clé publique)
- On prouvera que nos solutions marchent (sous certaines hypothèses)

# Crypto - organisation

- 2 groupes indépendants (A et B) , 2 enseignants (Roberto Amadio, Eugene Asarin), 95% même programme, même examen.
- Seulement un C/TD par semaine, mais travail systématique hors cours (« TD » et « TP » ) requis.
- DM à rendre contient des TPs sur la totalité du programme. À programmer en équipe dans votre langage préféré (Python, Java...).
- Examen fortement inspiré par les TD/TP
- Moodle avec supports détaillés (regardez-le, il n'est pas verrouillé)

# Pré-requis

- Notions basiques de *(mais on rappelle tout)*:
  - probabilités (discrètes)
  - arithmétique (modulaire, nombres premiers etc)
  - algèbre (groupes, anneaux)
  - algorithmique/complexité d'algo
- Pas d'allergie aux raisonnements mathématiques
- Certaine maîtrise de programmation (*partie TP 100% autonome*).

*Suivre la Sécurité en même temps conseillé mais pas indispensable*



# Machines virtuelles

P. Habermehl

## Machines Virtuelles

- Ce n'est pas un cours de machines virtuelles systèmes (VMWare, etc.)
- On parlera de machines virtuelles pour les programmes
- Exemples: MV OCAML, MV Java
- Code octet: instructions simples (mais encore compréhensibles)
- Comment fonctionne une telle machine ?
- Comment produire du code octet ?

# Machines virtuelles

P. Habermehl

## Machines Virtuelles

- Cours/TP
- Prérequis: OCAML (On va programmer en OCAML), Java
- Deux contrôles sur machine, un examen sur papier
- Utile si on veut faire le Master LP

# Programmation Web

P. Habermehl

## Programmation Web

- Rappel de HTML (contenu) et CSS (apparence)
- Apprentissage de Javascript
- Dynamiser le contenu des pages Web
- “Framework” pour le “Front”: jquery
- “Framework” pour le “Back”: node.js, express
- Connection avec base de données

# Programmation Web

P. Habermehl

## Programmation Web

- Adaptation aux différentes tailles d'écran: Bootstrap
- Cours/TP (plus de TP que cours)
- Prérequis: Aucun
- Evaluation: TP noté + projet (binôme)
- Utile pour Master GENIAL

# Langages de scripts:

S. Laplante & F. Laroussinie

## Python !

Est-ce vraiment un langage pourri ?

(Eviter d'écrire des trucs trop pourris,  
Python pour programmer des scripts,  
Python pour programmer des prototypes...)

Les structures de base du langage, les expressions régulières, les modules pour le système, les exceptions, les classes...

# Programmation ~~efficace~~ Programmation *rapide*

F. Laroussinie et D. Poulalhon

Un problème à résoudre en 2h.  
Par **équipe** de 5.

Obj.: Programmer rapidement.

- ▶ Certains problèmes sont difficiles et on cherche des solutions approchées (façon « google hashcode »).
- ▶ Certains problèmes peuvent être résolus complètement (façon « swerc »)

Langages suggérés: Java, C, Python,...

Appétence pour la programmation et l'algorithmique.

Travailler en équipe, venir à tous les CTP.

# Génie logiciel

E. Brignon

## Introduction au génie Logiciel

Description du cours

Emmanuel Bigeon

Institut de Recherche en Informatique Fondamentale  
Université Paris Diderot

2024

# Génie logiciel

E. Brignon

## Objectifs et Contenu

### Objectif

- Appréhender les besoins et les solutions au niveau des méthodes techniques et outils dédiés à la conception, développement, maintenance et opération des systèmes informatiques.
- Acquérir des compétence en modélisation de logiciel.

### Contenu

- Architecture Logicielle
  - Qualité de logiciel
  - Modélisation (UML)
    - Cahier des charges, cas d'utilisation
    - Organisation du logiciel
    - Comportement du logiciel
- Gestion de projet
- DevOps



# Génie logiciel

E. Brignon

## Objectifs et Contenu

### Objectif

- Appréhender les besoins et les solutions au niveau des méthodes techniques et outils dédiés à la conception, développement, maintenance et opération des systèmes informatiques.
- Acquérir des compétence en modélisation de logiciel.

### Contenu

- Architecture Logicielle
  - Qualité de logiciel
  - Modélisation (UML)
    - Cahier des charges, cas d'utilisation
    - Organisation du logiciel
    - Comportement du logiciel
- Gestion de projet
- DevOps

# Génie logiciel

E. Brignon

## Requis, Perspectives

### Pré-requis

- Intérêt pour le développement logiciel
- Programmation (langage Java)

### Cours précédents pouvant aider

- Conduite de Projet
- Programmation Orientée Objet
- Compléments de Programmation Orientée Objet

### Perspectives

- Développeur, architecte logiciel, gestionnaire de projet
- Poursuite en master informatique
  - GENIAL
  - LP
  - ...

# Sécurité informatique

P-E. Dagand

## Apprentissage théorique

- ▶ Représentation machine des programmes
- ▶ Pile d'appel et flot de contrôle
- ▶ Attaques sur le Web
- ▶ Fuites d'information

## Apprentissage pratique

1. Bombes logiques
2. Dépassements de tampon
3. Injections SQL, XSS

# Sécurité informatique

P-E. Dagand

## Pré-requis

- ▶ Cours Langage C du L2
  - ▶ Utilisation du compilateur C
  - ▶ Modèle mémoire C (pointeurs)
  - ▶ Convention d'appel de fonctions
- ▶ Cours Système d'exploitation du L3
  - ▶ Rôle du système d'exploitation
  - ▶ Maîtrise du shell & terminal Linux
  - ▶ Notion de processus

## Évaluation

- ▶ Glossaires (5%)
- ▶ Quiz (10%)
- ▶ TP, par binôme (35%)
- ▶ Examen final (50%)

# Bioinformatique

J.C. Gelly et J. Diharce

Premier CM: mardi 23 janvier !!

# Bioinformatique

J.C. Gelly et J. Diharce

" I think the biggest innovations of the 21st century will be at the intersection of **biology** and technology."  
"

Steve Jobs biography

# Bioinformatique

J.C. Gelly et J. Diharce

## Objectif de l'enseignement

- Sensibiliser et présenter les concepts fondamentaux de la bio-informatique
- Enseignements orientés « pratique »
- Nécessite un effort particulier du public informaticien

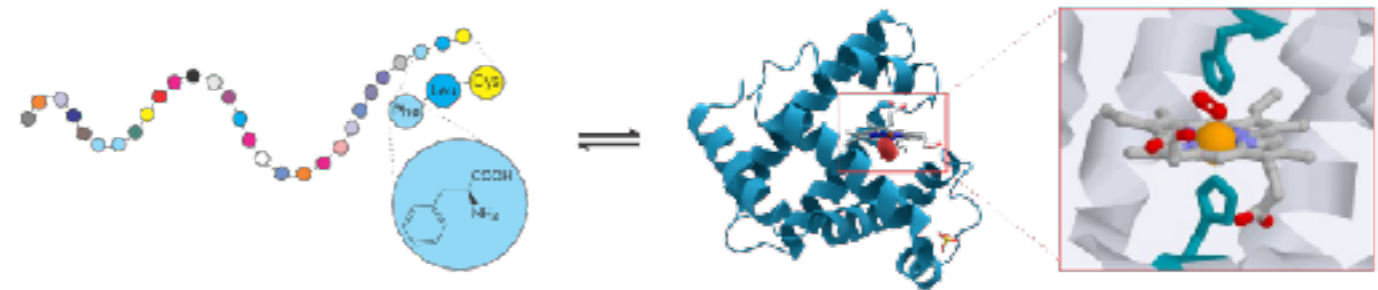
# Bioinformatique

J.C. Gelly et J. Diharce

## Concepts abordés

- Représentation des objets conceptuels de la biologie:

- Séquence biologique
- Structure moléculaire
- Réseaux biologique



*Credits: Zhang's lab*

- Systèmes d'information en biologie
- Méthodes et outils pour l'analyse des objets biologiques (séquence, structure, réseaux)



# Bioinformatique

J.C. Gelly et J. Diharce

## Les cours

- **1- mardi 23 janvier:** Entrepôt de données et Banques de données en biologie
- 2- mardi 30 janvier: Les méthodes 'omiques'
- 3- mardi 6 février: Introduction à la bio-informatique
- 4- mardi 13 février: La biologie systémique
- 5- mardi 20 février: Réseaux biologiques
- 6- mardi 27 février: Concepts de séquence-matrice-alignement-score
- 7- mardi 5 mars: Recherche de séquences avec BLAST, alignements multiples
- 8- mardi 12 mars: Bioinformatique structurale

# Travaux Dirigés et Pratiques

- TD1: Rappels de biologie pour les L3 info, mardi 13 février
- TD2 : Fouille de bases de données (Uniprot et Gene Ontology), mardi 27 février
- TD3: Alignement de séquences biologiques, mardi 12 mars
- TD4: Bio-informatique structurale et visualisation de macromolécules biologiques  
mardi 26 mars
- Travaux Pratiques, 4h, mardi 2 avril