

# Organisation du S6

L3 informatique  
11 décembre 2024



UFR d'Informatique

## Le semestre 6

Début des CM: semaine du 20 janvier  
Début des TD/TP: semaine du 27 janvier

Début des options: semaine du 27 janvier  
(sauf cas particuliers)

## 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>

## L'UE d'anglais

Mercredi:

- 14h30-16h30
- 16h30-18h30

Inscription: informations bientôt disponibles !

## 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 + 36h TD
Bioinformatique	3	24 h CTP
Linguistique	3	24 h CTP

(9 ects à choisir...)

(et un stage 6 ects)

## 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 devrait ouvrir rapidement.

**Vous pouvez d'ores et déjà voir les EDT et les compatibilités d'options.**

**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 les 6 et 7 janvier.

## Procédure

Lorsque le serveur sera ouvert pour le choix...

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

## Introduction au Génie Logiciel

Laetitia Laversa

### Objectif et contenu du cours

- ▶ Objectif du cours
  - ▶ Formaliser les grandes phases du développement logiciel (besoins, conception, développement et tests).
  - ▶ Être initié à l'écriture de tests unitaires et d'intégration.
- ▶ Contenu
  - ▶ Modélisation UML
  - ▶ Qualité logicielle
  - ▶ Tests
  - ▶ Gestion de projet

## Prérequis, Perspectives et Évaluation

- ▶ Pré-requis
  - ▶ Intérêt pour le développement logiciel
  - ▶ Programmation Java
- ▶ Perspectives
  - ▶ Développeur, architecte logiciel, gestionnaire de projet...
- ▶ Évaluation
  - ▶ Quizz à chaque TD (60%)
  - ▶ Contrôle final (40%)
  - ▶ Projet optionnel (20%)

## Machines virtuelles

Peter 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

- 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

# Sécurité informatique

Guillaume Geoffroy

## IF16Y100 : Sécurité Informatique

### Apprentissage théorique

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

### Apprentissage pratique

1. Analyse de programmes compilés
2. Exploitation de dépassements de tampon
3. Exploitation de vulnérabilités web

## IF16Y100 : Sécurité Informatique

### 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
  - ▶ Notion d'appel système

# Programmation web

Lélia Blin

## Programmation Web

Lélia Blin

### 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

Lélia Blin

### Programmation Web

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

## Probabilités

N. Cunéo

CM+TD : 6 ects

Premier CM: semaine du 20 janvier

TD: semaine du 27 janvier

## Bioinformatique

Jean-Christophe Gelly & Costas Bouyioukos

# Option Bio-informatique

Responsables : Jean-Christophe Gelly & Julien Diharce  
Intervenant : Costas Bouyioukos

“And Now for Something Completely Different”

Monty Pythons

***Biology is an Information science***

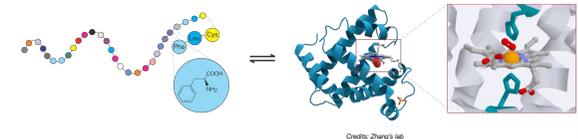
*IBM Research*

## Objectif de l’enseignement

- Sensibiliser et présenter les concepts fondamentaux de la bio-informatique
- Enseignements orientés « pratique », TD-TPs “Hands-on”
- Nécessite un effort particulier du public informaticien

## Concepts abordés

- Représentation des objets conceptuels de la biologie:
  - Séquence biologique
  - Structure moléculaire
  - Réseaux biologique
- Systèmes d’information en biologie
- Méthodes et outils informatique/statistique pour l’analyse des objets biologiques (séquence, structure, réseaux)





## 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

Roberto 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).

## Objectifs du cours

1. Définir les propriétés de **confidentialité** et d'**intégrité** :
  - **confidentialité** : le message chiffré “ne donne pas d’information”.
  - **intégrité** : le message “n’a pas été modifié/forgé”
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.

## Organisation

- 12 séances.
- **Cours** :
  - Diapos + poly sur la **page MOODLE** du cours.
  - Séance = cours+proposition exercices+discussion exercices proposés.
- **MCCC : Contrôle Continu Intégral**
  - **Présence régulière** aux séances nécessaire (sauf dispense).
  - **Travail en équipe** sur certains TD/TP proposés ; typiquement, mise en oeuvre d’un protocole cryptographique.
  - **Examen écrit individuel** qui porte en priorité sur les TD/TP proposés.

## Langages de scripts

Sophie Laplante & François Laroussinie

## 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

François Laroussinie & Dominique Poulalhon

## 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

## Conclusion

Consultez les EDT et les options **tout de suite**.

Soyez prêt-e-s pour l'ouverture du serveur pédagogique.