

Journée ISN, 13 juin 2012.

Comment est construite notre licence d'informatique ?



L'enseignement de la programmation et de l'algorithmique en L1.

L'enseignement de la programmation et de l'algorithmique en L1.

- La maquette de L1 pour les étudiants de math-info compte 90h d'informatique au premier semestre et 120h au second semestre.

L'enseignement de la programmation et de l'algorithmique en L1.

- La maquette de L1 pour les étudiants de math-info compte 90h d'informatique au premier semestre et 120h au second semestre.
- Elle comprend aussi 30h de raisonnements mathématiques (récurrence), 30h de suites et fonctions numériques, 66h d'algèbre linéaire (espaces vectoriels, matrices, systèmes linéaires).

L'enseignement de la programmation et de l'algorithmique en L1.

- La maquette de L1 pour les étudiants de math-info compte **90h d'informatique** au premier semestre et **120h** au second semestre.
- Elle comprend aussi 30h de raisonnements mathématiques (**réurrence**), 30h de **suites** et fonctions numériques, 66h d'algèbre linéaire (espaces vectoriels, **matrices**, systèmes linéaires).
- En informatique, il y a 9h de présentation de la matière, 81h de bases de l'informatique, **90h d'algorithmique et programmation** et 30h de programmation Web.

L'enseignement de la programmation et de l'algorithmique en L1.

- La maquette de L1 pour les étudiants de math-info compte **90h d'informatique** au premier semestre et **120h** au second semestre.
- Elle comprend aussi 30h de raisonnements mathématiques (**réurrence**), 30h de **suites** et fonctions numériques, 66h d'algèbre linéaire (espaces vectoriels, **matrices**, systèmes linéaires).
- En informatique, il y a 9h de présentation de la matière, 81h de bases de l'informatique, **90h d'algorithmique et programmation** et 30h de programmation Web.
- Les bases regroupent le **matériel**, la représentation des données, la **logique booléenne**, les bases de l'algorithmique (complexité) et **l'utilisation d'un système** à partir d'un langage de commande Shell.

L'enseignement de la programmation et de l'algorithmique en L1.

- La maquette de L1 pour les étudiants de math-info compte **90h d'informatique** au premier semestre et **120h** au second semestre.
- Elle comprend aussi 30h de raisonnements mathématiques (**réurrence**), 30h de **suites** et fonctions numériques, 66h d'algèbre linéaire (espaces vectoriels, **matrices**, systèmes linéaires).
- En informatique, il y a 9h de présentation de la matière, 81h de bases de l'informatique, **90h d'algorithmique et programmation** et 30h de programmation Web.
- Les bases regroupent le **matériel**, la représentation des données, la **logique booléenne**, les bases de l'algorithmique (complexité) et **l'utilisation d'un système** à partir d'un langage de commande Shell.
- **L'algorithmique et la programmation** se décomposent en 30h de bases de la programmation (structures de contrôle et types élémentaires) suivis de 60h d'algorithmique et programmation python.

L'enseignement de la programmation et de l'algorithmique en L1.

- La maquette de L1 pour les étudiants de math-info compte 90h d'informatique au premier semestre et 120h au second semestre.
- Elle comprend aussi 30h de raisonnements mathématiques (récurrence), 30h de suites et fonctions numériques, 66h d'algèbre linéaire (espaces vectoriels, matrices, systèmes linéaires).
- En informatique, il y a 9h de présentation de la matière, 81h de bases de l'informatique, 90h d'algorithmique et programmation et 30h de programmation Web.
- Les bases regroupent le matériel, la représentation des données, la logique booléenne, les bases de l'algorithmique (complexité) et l'utilisation d'un système à partir d'un langage de commande Shell.
- L'algorithmique et la programmation se décomposent en 30h de bases de la programmation (structures de contrôle et types élémentaires) suivis de 60h d'algorithmique et programmation python.
- L'enseignement de la programmation se fait en python, langage interprété inspiré de C.

La présentation de l'informatique aux étudiants de science.

La présentation de l'informatique aux étudiants de science.

- 1,5h de cours, 3h de TD et 4,5h de TP, autour du thème de [la cryptographie](#).

La présentation de l'informatique aux étudiants de science.

- 1,5h de cours, 3h de TD et 4,5h de TP, autour du thème de [la cryptographie](#).
- Principe de la [découverte scientifique](#) : de l'observation en TP à la théorie en cours.

La présentation de l'informatique aux étudiants de science.

- 1,5h de cours, 3h de TD et 4,5h de TP, autour du thème de [la cryptographie](#).
- Principe de la [découverte scientifique](#) : de l'observation en TP à la théorie en cours.
- Premier TP : [la programmation](#) (environnement python, calculette, structures conditionnelles, structures répétitives).

La présentation de l'informatique aux étudiants de science.

- 1,5h de cours, 3h de TD et 4,5h de TP, autour du thème de [la cryptographie](#).
- Principe de la [découverte scientifique](#) : de l'observation en TP à la théorie en cours.
- Premier TP : [la programmation](#) (environnement python, calculette, structures conditionnelles, structures répétitives).
- Second TP : [l'algorithmique](#) (cryptanalyse du chiffre de Vignère).

La présentation de l'informatique aux étudiants de science.

- 1,5h de cours, 3h de TD et 4,5h de TP, autour du thème de **la cryptographie**.
- Principe de la **découverte scientifique** : de l'observation en TP à la théorie en cours.
- Premier TP : **la programmation** (environnement python, calculette, structures conditionnelles, structures répétitives).
- Second TP : **l'algorithmique** (cryptanalyse du chiffre de Vignère).
- Troisième TP : **le matériel** (la carte à puce, communication, logiciel embarqué, carte bancaire et protocole cryptographique de paiement).

La présentation de l'informatique aux étudiants de science.

- 1,5h de cours, 3h de TD et 4,5h de TP, autour du thème de [la cryptographie](#).
- Principe de la [découverte scientifique](#) : de l'observation en TP à la théorie en cours.
- Premier TP : [la programmation](#) (environnement python, calculette, structures conditionnelles, structures répétitives).
- Second TP : [l'algorithmique](#) (cryptanalyse du chiffre de Vignère).
- Troisième TP : [le matériel](#) (la carte à puce, communication, logiciel embarqué, carte bancaire et protocole cryptographique de paiement).
- Premier TD : [la représentation de l'information](#) (systèmes de numération, codage des nombres dans un ordinateur, images, textes).

La présentation de l'informatique aux étudiants de science.

- 1,5h de cours, 3h de TD et 4,5h de TP, autour du thème de [la cryptographie](#).
- Principe de la [découverte scientifique](#) : de l'observation en TP à la théorie en cours.
- Premier TP : [la programmation](#) (environnement python, calculette, structures conditionnelles, structures répétitives).
- Second TP : [l'algorithmique](#) (cryptanalyse du chiffre de Vignère).
- Troisième TP : [le matériel](#) (la carte à puce, communication, logiciel embarqué, carte bancaire et protocole cryptographique de paiement).
- Premier TD : [la représentation de l'information](#) (systèmes de numération, codage des nombres dans un ordinateur, images, textes).
- Second TD : [les codes correcteurs](#) (code de Hamming, compression sans perte).

La présentation de l'informatique aux étudiants de science.

- 1,5h de cours, 3h de TD et 4,5h de TP, autour du thème de [la cryptographie](#).
- Principe de la [découverte scientifique](#) : de l'observation en TP à la théorie en cours.
- Premier TP : [la programmation](#) (environnement python, calculette, structures conditionnelles, structures répétitives).
- Second TP : [l'algorithmique](#) (cryptanalyse du chiffre de Vignère).
- Troisième TP : [le matériel](#) (la carte à puce, communication, logiciel embarqué, carte bancaire et protocole cryptographique de paiement).
- Premier TD : [la représentation de l'information](#) (systèmes de numération, codage des nombres dans un ordinateur, images, textes).
- Second TD : [les codes correcteurs](#) (code de Hamming, compression sans perte).
- Cours : comment passer [d'un problème au calcul](#) de sa solution par ordinateur en passant par un algorithme et un programme.

Notre licence.

Notre licence.

- Le socle qui nous est nécessaire est [la programmation](#) adossée à [l'algorithmique](#). C'est enseigné en L1 en 180h.

Notre licence.

- Le socle qui nous est nécessaire est **la programmation** adossée à **l'algorithmique**. C'est enseigné en L1 en 180h.
- Sur ce socle acquis en L1, on bâtit un ensemble regroupant la compilation, le système (son organisation) et l'architecture pour expliquer **comment un programme s'exécute sur le matériel**. C'est enseigné en L2 et L3 en 210h.

Notre licence.

- Le socle qui nous est nécessaire est **la programmation** adossée à **l'algorithmique**. C'est enseigné en L1 en 180h.
- Sur ce socle acquis en L1, on bâtit un ensemble regroupant la compilation, le système (son organisation) et l'architecture pour expliquer **comment un programme s'exécute sur le matériel**. C'est enseigné en L2 et L3 en 210h.
- A cet ensemble, on ajoute en L2 : **réseaux (60h)**, **programmation Web (30h)**, calcul scientifique (40h), **programmation de robots (70h)**, contribution à un projet opensource (40h).

Notre licence.

- Le socle qui nous est nécessaire est la **programmation** adossée à **l'algorithmique**. C'est enseigné en L1 en 180h.
- Sur ce socle acquis en L1, on bâtit un ensemble regroupant la compilation, le système (son organisation) et l'architecture pour expliquer **comment un programme s'exécute sur le matériel**. C'est enseigné en L2 et L3 en 210h.
- A cet ensemble, on ajoute en L2 : **réseaux (60h)**, **programmation Web (30h)**, calcul scientifique (40h), **programmation de robots (70h)**, contribution à un projet opensource (40h).
- En L3 : **base de données et Web (67h)**, **algorithmique et complexité (40h)**, poo (40h), théorie de l'information (40h), arithmétique des ordinateurs (60h), interfaces graphiques (30h), programmation parallèle (50h), programmation fonctionnelle (30h), algorithmique stochastique (30h).

Remarques d'ordre pédagogique concernant notre licence.

Remarques d'ordre pédagogique concernant notre licence.

- Le choix d'un **environnement interprété** pose des problèmes.

Remarques d'ordre pédagogique concernant notre licence.

- Le choix d'un **environnement interprété** pose des problèmes.
- L'environnement **ne sépare pas les phases de rédaction et d'exécution**. On peut retoucher un texte et le ré-exécuter. Or, les exécutions passées influent sur l'exécution courante.

Remarques d'ordre pédagogique concernant notre licence.

- Le choix d'un **environnement interprété** pose des problèmes.
- L'environnement **ne sépare pas les phases de rédaction et d'exécution**. On peut retoucher un texte et le ré-exécuter. Or, les exécutions passées influent sur l'exécution courante.
- Il faut **se plier à une discipline** qui revient à réinitialiser l'interpréteur avant chaque exécution, ce qui limite son intérêt.

Remarques d'ordre pédagogique concernant notre licence.

Remarques d'ordre pédagogique concernant notre licence.

- La phase d'apprentissage de la programmation élémentaire est assez longue (entre 15 et 30 heures selon les élèves pour commencer à savoir écrire un programme contenant des affectations, des tests et des boucles simples).

Remarques d'ordre pédagogique concernant notre licence.

- La phase d'apprentissage de la programmation élémentaire est assez longue (entre 15 et 30 heures selon les élèves pour commencer à savoir écrire un programme contenant des affectations, des tests et des boucles simples).
- C'est une fois cette première phase acquise qu'on aborde l'algorithmique. Ils ont alors un vocabulaire de base (affectation, test, boucle, variable, vecteur) permettant d'exprimer quelques méthodes et algorithmes simples (recherche du plus petit élément, insertion en place, gestion d'une pile ou d'une file en table, ...).

Remarques d'ordre pédagogique concernant notre licence.

- La phase d'apprentissage de la programmation élémentaire est assez longue (entre 15 et 30 heures selon les élèves pour commencer à savoir écrire un programme contenant des affectations, des tests et des boucles simples).
- C'est une fois cette première phase acquise qu'on aborde l'algorithmique. Ils ont alors un vocabulaire de base (affectation, test, boucle, variable, vecteur) permettant d'exprimer quelques méthodes et algorithmes simples (recherche du plus petit élément, insertion en place, gestion d'une pile ou d'une file en table, ...).
- L'algorithmique vise deux objectifs : la vérification (preuve) et la performance (complexité). En première année de licence, on étudie divers algorithmes de tri, de recherche et quelques structures de données (pile et file) comme types abstraits.

Remarques d'ordre pédagogique concernant notre licence.

- La phase d'apprentissage de la programmation élémentaire est assez longue (entre 15 et 30 heures selon les élèves pour commencer à savoir écrire un programme contenant des affectations, des tests et des boucles simples).
- C'est une fois cette première phase acquise qu'on aborde l'algorithmique. Ils ont alors un vocabulaire de base (affectation, test, boucle, variable, vecteur) permettant d'exprimer quelques méthodes et algorithmes simples (recherche du plus petit élément, insertion en place, gestion d'une pile ou d'une file en table, ...).
- L'algorithmique vise deux objectifs : la vérification (preuve) et la performance (complexité). En première année de licence, on étudie divers algorithmes de tri, de recherche et quelques structures de données (pile et file) comme types abstraits.
- L'enseignement de l'algorithmique est perçu, par les élèves de L1, comme relativement abstrait et décorrélé de la programmation ("à quoi ça sert de prouver ? à quoi ça sert d'être en $n \cdot \log(n)$?").

Remarques d'ordre pédagogique concernant notre licence.

- La phase d'apprentissage de la programmation élémentaire est assez longue (entre 15 et 30 heures selon les élèves pour commencer à savoir écrire un programme contenant des affectations, des tests et des boucles simples).
- C'est une fois cette première phase acquise qu'on aborde l'algorithmique. Ils ont alors un vocabulaire de base (affectation, test, boucle, variable, vecteur) permettant d'exprimer quelques méthodes et algorithmes simples (recherche du plus petit élément, insertion en place, gestion d'une pile ou d'une file en table, ...).
- L'algorithmique vise deux objectifs : la vérification (preuve) et la performance (complexité). En première année de licence, on étudie divers algorithmes de tri, de recherche et quelques structures de données (pile et file) comme types abstraits.
- L'enseignement de l'algorithmique est perçu, par les élèves de L1, comme relativement abstrait et décorrélé de la programmation ("à quoi ça sert de prouver ? à quoi ça sert d'être en $n \cdot \log(n)$?").
- C'est la suite, en L2 et L3, qui leur permet de mesurer le bénéfice d'une démarche algorithmique sur une démarche d'écriture directe d'un programme.