

Science informatique

Objectifs

L'enseignement de l'informatique a pour but de faire découvrir et d'acquérir les notions de base concernant la représentation (information et données), l'organisation (réseaux, bases de données) et le traitement automatique des données (algorithmique et programmation). Il encourage l'élève à faire un usage responsable des technologies numériques (bonnes pratiques, sécurité). Les savoir-faire de la science informatique permettent de développer différents schémas de résolution de problèmes et leurs mises en œuvre, favorisant différentes compétences universelles comme la créativité de l'élève ou sa capacité à raisonner de manière structurée et à travailler avec précision. Les exemples sont proposés de manière indicative.

L'enseignement se déroule essentiellement au travers de travaux pratiques individuels ou en groupe, ou sous forme d'ateliers, pour inciter les élèves à évaluer de façon autonome et immédiate la qualité de leur analyse et de leurs modélisations.

Le nombre de séances associé aux thèmes apparaît également de manière indicative, de façon à structurer le temps qui leur est consacré.

Évaluation

Si l'évaluation peut se faire à partir de travaux écrits, il est souhaitable de privilégier travaux pratiques et mise en œuvre de projets (particulièrement en 2^e année).

Thèmes	Objectifs	Notions et concepts	Savoir-faire	Exemples	Séances
INFORMATION ET DONNEES					
• Représentation numérique de l'information	• Comprendre que l'ordinateur encode de différentes manières et distinguer les avantages et les inconvénients de ces encodages	• Discrétisation	• Savoir encoder de l'information	• Encoder un nombre ou un caractère	2
• Ordinateur / machine physique	• Comprendre les caractéristiques qui décrivent une machine	• Architecture d'un ordinateur	• Identifier le rôle des composants principaux d'un ordinateur	• Lire et comprendre un catalogue de vente	2
			• Distinguer les périphériques d'entrée, de sortie et mixtes		
		• Types de mesure	• Comparer la puissance de deux machines • Comparer des ordres de grandeur	• Hz, Octet, Watt	
• Stockage de données	• Avoir conscience de la problématique de gestion des données	• Support local ou distant des données	• Gérer ses données	• Utiliser un service de stockage distant	2
		• Pérennité du stockage et de l'accès aux données	• Être sensible à l'évolution et à l'obsolescence des supports et des standards	• Montrer différents types de supports	
• Impact écologique de l'informatique	• Avoir conscience du coût écologique du stockage de données			• Discuter du coût énergétique de l'informatique	2

Thèmes	Objectifs	Notions et concepts	Savoir-faire	Exemples	Séances
ALGORITHMIQUE ET PROGRAMMATION					
• Structures de données	• Extraire les données d'un problème	• Types scalaires	• Savoir extraire les données d'un problème	• Établir une liste d'élèves	8
		• Listes, arbres et graphes			
		• Expressions	• Comprendre qu'une expression se réduit à une valeur		
	• Comprendre que les données sont associées à des opérations	• Typage	• Connaître les opérations associées aux types scalaires et composés	• Reconnaître une application	
• Concepts génériques de résolution de problèmes	• Comprendre que de nombreuses activités humaines impliquent une procédure algorithmique	• Algorithme	• Établir une procédure systématique	• Transmettre une recette de cuisine	4
	• Subdiviser un problème en sous-problèmes	• Fonction	• Comprendre un algorithme de tri	• Discuter de l'implémentation d'un algorithme de tri	
• Principes de programmation	• Utiliser des structures de données	• Types scalaires	• Savoir différencier des types simples	• Utiliser des opérations arithmétiques	30
		• Listes	• Savoir opérer sur des listes	• Concaténer deux listes	
	• Utiliser des variables	• Association nom/valeur	• Savoir affecter une variable et utiliser sa valeur	• Permuter des données	
			• Savoir reconnaître le type d'une variable		
			• Savoir différencier une variable d'une constante		

Thèmes	Objectifs	Notions et concepts	Savoir-faire	Exemples	Séances
ALGORITHMIQUE ET PROGRAMMATION (suite)					
• Principes de programmation (suite)	• Utiliser des structures de contrôle	• Séquences, tests, boucles	• Savoir implémenter un test	• Implémenter le calcul du maximum d'une paire de nombres	
			• Savoir implémenter une boucle	• Extraire un élément d'une liste	
	• Déterminer le résultat d'un programme	• Syntaxe et sémantique		• Corriger un programme	
	• Définir et utiliser des fonctions	• Fonction	• Factoriser du code	• Décomposer un problème en plusieurs étapes indépendantes	
RESEAUX					
• Réseaux / organisation et protocoles	• Comprendre les principes d'organisation d'un réseau informatique et l'utilité des protocoles de communication	• Transmission, perte, corruption et respect de l'intégrité de l'information	• Comprendre le principe de couches d'abstraction	• Illustrer le cheminement d'une requête sur Internet	8
		• Transport, routage, adresse, protocole	• Savoir distinguer une adresse IP d'une adresse URL	• Transmettre de manière débranchée une information en classe	
		• Client/serveur	• Différencier le Web de l'Internet	• Expérimenter le modèle sur un service Web	
SECURITE					
• Sécurité	• Connaître les aspects pratiques relatifs à la sécurité		• Connaître différents types d'attaques (moyens) et leurs parades	• Étudier le fonctionnement d'un déni de service	2
			• Distinguer les objectifs d'attaques cybernétiques	• Étudier une attaque historique relatée par la presse	

Thèmes	Objectifs	Notions et concepts	Savoir-faire	Exemples	Séances
ALGORITHMIQUE ET PROGRAMMATION					
• Concepts algorithmiques	• Connaître différentes stratégies algorithmiques	• Stratégie gloutonne, diviser pour régner	• Mettre en œuvre un algorithme en se basant sur une stratégie algorithmique, sur la base des connaissances approchées en première	• Calculer la moyenne de l'âge des personnes dont le nom commence par A • Classer par ordre alphabétique les personnes dont le nom commence par A	8
	• Approcher un algorithme (sans nécessairement l'implémenter)		• Expliquer au moins un algorithme	• Expliquer une version simplifiée de l'algorithme de Dijkstra	
	• En partant d'une étude débranchée d'un problème, l'implémenter				
• Principes de programmation (approfondir les notions approchées en première année)	• Comprendre la portée d'une variable	• Portée lexicale et dynamique	• Identifier la portée d'une variable	• Déterminer les points d'un programme où une variable est visible	6
	• Aborder le principe de récursion	• Récursion	• Écrire une fonction récursive	• Implémenter la recherche dans une liste de manière récursive	
• Débogage	• Aborder les notions de débogage	• Évaluation pas-à-pas	• Identifier des erreurs de syntaxe et de sémantique	• Corriger la condition de sortie d'une boucle	2
SECURITE					
• Sécurité	• Aborder les notions théoriques de disponibilité, d'intégrité, de confidentialité, de non-répudiation et d'authentification ; sensibiliser à la sécurité liée à l'échange d'informations	• Chiffrement	• Comprendre et expérimenter le fonctionnement d'un algorithme de chiffrement, ses forces et ses faiblesses	• Expérimenter le chiffre de César	4
		• Transfert d'information chiffrée	• Comprendre l'importance des protocoles de transmission et faire le lien avec les protocoles réseaux	• Distinguer les systèmes à clés symétriques et asymétriques	

Thèmes	Objectifs	Notions et concepts	Savoir-faire	Exemples	Séances
BASES DE DONNEES					
• Bases de données	• Décrire un système de gestion de bases de données	• Organisation des données	• Observer une base de données pour comprendre sa structure	• Examiner une base de données existante	10
		• Extraction des données	• Effectuer des commandes élémentaires de requête	• Extraire le classement par catégorie d'une course	
			• Sensibiliser à l'usage statistique des données et à leurs impacts sociétaux	• Examiner les usages politiques et économiques de la collecte de données	

Savoir-faire transversaux à exercer à toute occasion

Citoyenneté numérique

- Prendre conscience des droits et des devoirs de l'utilisateur des technologies numériques

Bonnes pratiques concernant l'usage d'un système d'exploitation

- Connaître le nom de plusieurs systèmes d'exploitation et les appareils sur lesquels ils sont installés
- Comprendre les liens entre utilisateurs, périphériques, applications et systèmes d'exploitation
- Savoir que le format de stockage dépend du support et du système d'exploitation
- Comprendre la différence entre un fichier et un dossier
- Distinguer différents types de fichiers
- Maîtriser la notion d'arborescence de fichiers à travers quelques exemples
- Savoir installer, mettre à jour, supprimer, protéger des applications
- Constituer un mot de passe efficace

Bonnes pratiques concernant la gestion des données

- Savoir nommer correctement des fichiers et des dossiers
- Savoir organiser correctement des fichiers et des dossiers
- Savoir protéger ses données en mettant en place différents types de sauvegarde
- Savoir compresser des données
- Savoir synchroniser des données par le biais d'outils ou de logiciels appropriés
- Savoir partager des données

Sensibilisation à différents aspects sécuritaires

- Adopter une attitude réfléchie et responsable
- Prendre conscience que toute activité numérique laisse des traces