

Progression première NSI - Année 2021-2022

Chapitre	Contenus	Capacités attendues à la fin du chapitre
1. Types et valeurs de base 1	<ol style="list-style-type: none"> 1. Écriture d'un entier positif dans une base b 2. Évaluation du nombre de bits nécessaires 3. Représentation binaire d'un entier relatif 4. Représentation approximative des nombres réels : notion de nombre flottant 	<p>Passer de la représentation d'une base dans une autre.</p> <p>Évaluer le nombre de bits nécessaires à l'écriture en base 2 d'un entier, de la somme ou du produit de deux nombres entiers.</p> <p>Utiliser le complément à 2.</p> <p>Calculer sur quelques exemples la représentation de nombres réels : 0.1, 0.25 ou 1/3.</p>
2. Types et valeurs de base 2	<ol style="list-style-type: none"> 1. Valeurs booléennes : 0 (Faux) ou 1 (Vrai). 2. Expressions booléennes 3. Représentation d'un texte en machine. 	<p>Dresser la table d'une expression booléenne.</p> <p>Identifier l'intérêt des différents systèmes d'encodage.</p> <p>Convertir un fichier texte dans différents formats d'encodage.</p>
3. Types construits	<ol style="list-style-type: none"> 1. p-uplets. p-uplets nommés 2. Tableau indexé, tableau donné en compréhension 3. Dictionnaires par clés et valeurs 	<p>Écrire une fonction renvoyant un p-uplet de valeurs.</p> <p>En Python, implémenter un p-uplet nommé par un dictionnaire.</p> <p>Lire et modifier les éléments d'un tableau grâce à leurs index.</p> <p>Construire un tableau par compréhension.</p> <p>Utiliser des tableaux de tableaux pour représenter des matrices : notation <code>a[i][j]</code>.</p> <p>Itérer sur les éléments d'un tableau.</p> <p>Construire une entrée de dictionnaire.</p> <p>Itérer sur les éléments d'un dictionnaire.</p> <p>Utiliser les méthodes <code>keys()</code>, <code>values()</code> et <code>items()</code>.</p>
4. Matériel et systèmes d'exploitation 1	<ol style="list-style-type: none"> 1. Architecture de réseaux 2. Les protocoles de communication 3. Interface Homme-Machine (IHM) : fonctions associées à des capteurs et des actionneurs 	<p>Mettre en évidence l'intérêt du découpage des données en paquets et de leur encapsulation.</p> <p>Dérouler le fonctionnement d'un protocole simple de récupération de perte de paquets (bit alterné).</p> <p>Simuler ou mettre en œuvre un réseau.</p> <p>Identifier le rôle des capteurs et actionneurs.</p> <p>Réaliser par programmation une IHM répondant à un cahier des charges donné.</p>
5. Données en tables	<ol style="list-style-type: none"> 1. Indexation de tables 2. Recherche dans une table 3. Tri d'une table 4. Fusion de tables 	<p>Importer une table depuis un fichier texte tabulé ou un fichier CSV.</p> <p>Rechercher les lignes d'une table vérifiant des critères exprimés en logique propositionnelle</p> <p>Trier une table suivant une colonne.</p> <p>Construire une nouvelle table en combinant les données de deux tables.</p>

<p>6. Interactions sur le web</p>	<ol style="list-style-type: none"> 1. Modalités de l'interaction entre l'homme et la machine 2. Interaction avec l'utilisateur d'une page Web 3. Interaction client-serveur ; Requêtes HTTP, réponses du serveur 4. Formulaire d'une page Web 	<p>Identifier les différents composants graphiques permettant d'interagir avec une application Web.</p> <p>Identifier les événements que les fonctions associées aux différents composants graphiques sont capables de traiter.</p> <p>Analyser et modifier les méthodes exécutées lors d'un clic sur un bouton d'une page Web.</p> <p>Distinguer ce qui est exécuté sur le client ou sur le serveur et dans quel ordre.</p> <p>Distinguer ce qui est mémorisé dans le client et retransmis au serveur.</p> <p>Reconnaître quand et pourquoi la transmission est chiffrée.</p> <p>Analyser le fonctionnement d'un formulaire simple.</p> <p>Distinguer les transmissions de paramètres par les requêtes POST ou GET.</p>
<p>7. Algorithmique 1</p>	<ol style="list-style-type: none"> 1. Parcours séquentiel d'un tableau 2. Tri par sélection, tri par insertion 	<p>Écrire un algorithme de recherche d'une occurrence sur des valeurs de type quelconque.</p> <p>Écrire un algorithme de recherche d'un extremum, de calcul d'une moyenne.</p> <p>On montre que le coût est linéaire.</p> <p>Écrire un algorithme de tri. Décrire un invariant de boucle qui prouve la correction des tris par insertion, par sélection.</p> <p>La terminaison de ces algorithmes est à justifier.</p> <p>On montre que leur coût est quadratique dans le pire cas.</p>
<p>8. Matériel et systèmes d'exploitation 2</p>	<ol style="list-style-type: none"> 1. Modèle d'architecture séquentielle (von Neumann) 2. Systèmes d'exploitation 	<p>Distinguer les rôles et les caractéristiques des différents constituants d'une machine.</p> <p>Dérouler l'exécution d'une séquence d'instructions simples du type langage machine.</p> <p>Identifier les fonctions d'un système d'exploitation.</p> <p>Utiliser les commandes de base en ligne de commande.</p> <p>Gérer les droits et permissions d'accès aux fichiers.</p>
<p>9. Langages et programmation</p>	<ol style="list-style-type: none"> 1. Constructions élémentaires 2. Diversité et unité des langages de programmation 3. Spécification 4. Mise au point des programmes 5. Utilisation de bibliothèques 	<p>Mettre en évidence un corpus de constructions élémentaires.</p> <p>Repérer, dans un nouveau langage de programmation, les traits communs et les traits particuliers à ce langage.</p> <p>Prototyper une fonction.</p> <p>Décrire les préconditions sur les arguments.</p> <p>Décrire des postconditions sur les résultats.</p> <p>Utiliser des jeux de tests.</p> <p>Utiliser la documentation d'une bibliothèque.</p>
<p>10. Algorithmique 2</p>	<ol style="list-style-type: none"> 1. Algorithme des k plus proches voisins 2. Recherche dichotomique dans un tableau trié 3. Algorithmes gloutons 	<p>Écrire un algorithme qui prédit la classe d'un élément en fonction de la classe majoritaire de ses k plus proches voisins.</p> <p>Montrer la terminaison de la recherche dichotomique à l'aide d'un variant de boucle.</p> <p>Résoudre un problème grâce à un algorithme glouton.</p>