| **Progression première NSI** - *Année 2023-2024* | | |
| --- | --- | --- |
| Chapitre | Contenus | Capacités attendues à la fin du chapitre |
| **1. Types et valeurs de base 1** | 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 | Passer de la représentation d’une base dans une autre.  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.  Utiliser le complément à 2.  Calculer sur quelques exemples la représentation de nombres réels : 0.1, 0.25 ou 1/3. |
| **2. Types et valeurs de base 2** | 1. Valeurs booléennes : 0 (Faux) ou 1 (Vrai). 2. Expressions booléennes 3. Représentation d’un texte en machine. | Dresser la table d’une expression booléenne.  Identifier l’intérêt des différents systèmes d’encodage.  Convertir un fichier texte dans différents formats d’encodage. |
| **3. Types construits** | 1. p-uplets. p-uplets nommés 2. Tableau indexé, tableau donné en compréhension 3. Dictionnaires par clés et valeurs | Écrire une fonction renvoyant un p-uplet de valeurs.  En Python, implémenter un p-uplet nommé par un dictionnaire.  Lire et modifier les éléments d’un tableau grâce à leurs index.  Construire un tableau par compréhension.  Utiliser des tableaux de tableaux pour représenter des matrices : notation a[i][j].  Itérer sur les éléments d’un tableau.  Construire une entrée de dictionnaire.  Itérer sur les éléments d’un dictionnaire.  Utiliser les méthodes keys(), values() et items(). |
| **4. Matériel et systèmes d'exploitation 1** | |  | | --- | | 1. Architecture de réseaux 2. Les protocoles de communication 3. Interface Homme-Machine (IHM) : fonctions associées à des capteurs et des actionneurs | | Mettre en évidence l’intérêt du découpage des données en paquets et de leur encapsulation.  Dérouler le fonctionnement d’un protocole simple de récupération de perte de paquets (bit alterné).  Simuler ou mettre en œuvre un réseau.  Identifier le rôle des capteurs et actionneurs.  Réaliser par programmation une IHM répondant à un cahier des charges donné. |
| **5. Données en tables** | 1. Indexation de tables 2. Recherche dans une table 3. Tri d’une table 4. Fusion de tables | Importer une table depuis un fichier texte tabulé ou un fichier CSV.  Rechercher les lignes d’une table vérifiant des critères exprimés en logique propositionnelle  Trier une table suivant une colonne.  Construire une nouvelle table en combinant les données de deux tables. |
| **6. Interactions sur le web** | 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 | Identifier les différents composants graphiques permettant d’interagir avec une application Web.  Identifier les événements que les fonctions associées aux différents composants graphiques sont capables de traiter.  Analyser et modifier les méthodes exécutées lors d’un clic sur un bouton d’une page Web.  Distinguer ce qui est exécuté sur le client ou sur le serveur et dans quel ordre.  Distinguer ce qui est mémorisé dans le client et retransmis au serveur.  Reconnaître quand et pourquoi la transmission est chiffrée.  Analyser le fonctionnement d’un formulaire simple.  Distinguer les transmissions de paramètres par les requêtes POST ou GET. |
| **7. Algorithmique 1** | 1. Parcours séquentiel d’un tableau 2. Tri par sélection, tri par insertion | Écrire un algorithme de recherche d’une occurrence sur des valeurs de type quelconque.  Écrire un algorithme de recherche d’un extremum, de calcul d’une moyenne.  On montre que le coût est linéaire.  Écrire un algorithme de tri. Décrire un invariant de boucle qui prouve la correction des tris par insertion, par sélection.  La terminaison de ces algorithmes est à justifier.  On montre que leur coût est quadratique dans le pire cas. |
| **8. Matériel et systèmes d'exploitation 2** | 1. Modèle d’architecture séquentielle (von Neumann) 2. Systèmes d’exploitation | Distinguer les rôles et les caractéristiques des différents constituants d’une machine.  Dérouler l’exécution d’une séquence d’instructions simples du type langage machine.  Identifier les fonctions d’un système d’exploitation.  Utiliser les commandes de base en ligne de commande.  Gérer les droits et permissions d’accès aux fichiers. |
| **9. Langages et programmation** | 1. Constructions élémentaires 2. Diversité et unité des langages de programmation 3. Spécification 4. Mise au point des programmes 5. Utilisation de bibliothèques | Mettre en évidence un corpus de constructions élémentaires.  Repérer, dans un nouveau langage de programmation, les traits communs et les traits particuliers à ce langage.  Prototyper une fonction.  Décrire les préconditions sur les arguments.  Décrire des postconditions sur les résultats.  Utiliser des jeux de tests.  Utiliser la documentation d’une bibliothèque. |
| **10. Algorithmique 2** | 1. Algorithme des k plus proches voisins 2. Recherche dichotomique dans un tableau trié 3. Algorithmes gloutons | Écrire un algorithme qui prédit la classe d’un élément en fonction de la classe majoritaire de ses k plus proches voisins.  Montrer la terminaison de la recherche dichotomique à l’aide d’un variant de boucle.  Résoudre un problème grâce à un algorithme glouton. |