

République Tunisienne  
Ministère de l'Éducation

A decorative graphic on the left side of the page consists of several overlapping, semi-transparent blue squares of various sizes. Some squares contain small white dots, and the overall effect is a dynamic, digital-looking pattern. The squares are arranged in a way that suggests movement and depth, with some appearing to be in the foreground and others receding into the background.

# Aides pédagogiques **D'INFORMATIQUE**

SECTIONS : SCIENCES EXPERIMENTALES,  
SCIENCES TECHNIQUES  
ET MATHEMATIQUES

Septembre 2024

## NIVEAU : 2<sup>EME</sup> ANNEE

### Aide pédagogique 2024-2025

Domaine d'apprentissage	Savoirs associés	Pistes pédagogiques et directives
<p style="text-align: center;"><b>Pensée Computationnelle et programmation</b></p>	<ul style="list-style-type: none"> <li>• <b>Identifier les étapes de résolution d'un problème.</b> <ul style="list-style-type: none"> <li>○ Dégager les éléments essentiels permettant la résolution d'un problème.</li> </ul> </li> <li>• <b>Élaborer des solutions sous forme d'algorithmes.</b> <ul style="list-style-type: none"> <li>○ Utiliser des structures de données à bon escient (<b>constantes</b> et <b>variables</b>)</li> <li>○ Utiliser les types de données : <b>entier, réel, caractère, booléen, chaîne de caractères</b></li> <li>○ Les structures simples (<b>Lecture, affichage et affectation</b>).</li> <li>○ Utiliser les structures de contrôle adéquates pour résoudre un problème.                             <ul style="list-style-type: none"> <li>▪ La structure de contrôle conditionnelle (<b>Si ... Alors ; Si ... Sinon</b>).</li> <li>▪ La structure de contrôle répétitive (<b>Pour</b>).</li> </ul> </li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Etablir des liens et trouver des fils conducteurs entre les différents domaines d'apprentissage rompant ainsi avec l'aspect linéaire du programme.</li> <li>• Il est préconisé de présenter les savoirs associés à travers des projets, des mini-projets ou des activités ayant du sens pour l'apprenant.</li> <li>• Favoriser l'investigation, le questionnement, l'apprentissage expérientiel, l'apprentissage par problème. etc.</li> <li>• Il est fortement recommandé d'opter pour une démarche de création au cours de laquelle les apprenants développent leur autonomie, leur créativité et leur imagination, mais aussi le sens du travail collaboratif.</li> <li>• Donner du sens aux activités, les diversifier et opter pour une démarche interdisciplinaire permettant le décloisonnement entre les divers champs d'apprentissages et l'ouverture de l'informatique sur les autres disciplines.</li> <li>• Favoriser l'exploitation des ressources en ligne et développer la communication.</li> <li>• Choisir des problèmes (activités) du vécu de l'élève</li> <li>• Inciter à l'innovation et motiver les apprenants pour la créativité.</li> <li>• Inciter les apprenants à faire des échanges autour des solutions proposées et de les partager en ligne.</li> <li>• Il est pertinent de :</li> </ul>

	<ul style="list-style-type: none"> <li>• <b>Exploiter un environnement de programmation</b> <ul style="list-style-type: none"> <li>○ Implémenter un algorithme en utilisant un langage de programmation.</li> <li>○ Tester et valider un programme.</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>○ familiariser l'apprenant à formuler sous forme d'actions des solutions à des problèmes puisés de son vécu. On pourra exprimer ces solutions à l'aide d'un schéma, d'un organigramme, d'une carte heuristique, d'un pseudocode, etc.</li> <li>○ inviter les apprenants à déterminer les entrées, les sorties et les traitements.</li> <li>○ inciter les apprenants à vérifier la validité d'une solution donnée par rapport à l'énoncé d'un problème.</li> <li>• Il est recommandé d'inciter les apprenants à : <ul style="list-style-type: none"> <li>○ étudier quelques séquences algorithmiques (décrire, comparer, déterminer le rôle, etc.).</li> <li>○ modifier des algorithmes existants pour changer leurs comportements.</li> <li>○ corriger les erreurs de logique dans une séquence algorithmique afin de parvenir aux résultats souhaités.</li> <li>○ évaluer différentes solutions algorithmiques d'un même problème donné.</li> </ul> </li> <li>• Il est préconisé : <ul style="list-style-type: none"> <li>○ de familiariser l'apprenant avec un environnement de programmation.</li> <li>○ d'inciter l'apprenant à : <ul style="list-style-type: none"> <li>▪ réutiliser des codes sources existants.</li> <li>▪ modifier un programme existant pour obtenir un résultat différent.</li> <li>▪ écrire un programme pour résoudre un problème.</li> </ul> </li> </ul> </li> <li>• Toutes les solutions des problèmes sont implémentées via le langage de programmation <b>Python</b>.</li> </ul>
--	--	---

Systemes et  
technologies  
Internet

- **Prendre conscience de l'intérêt de la robotique**

- Définir la robotique.
- Identifier des domaines d'application de la robotique.

- **Piloter un objet connecté**

- Connaître les interfaces de l'objet à connecter.
- Connecter un objet à l'ordinateur
- Programmer des objets simples virtuels ou réels pour réaliser différentes tâches d'une façon innovante.

- Il est nécessaire de présenter la robotique et d'expliquer ses fondements en s'appuyant sur des séquences vidéo, des ressources numériques, des études de cas, etc.
- L'apprenant n'est pas appelé à développer une application de commande.
- Utiliser Micro-Python / Arduino pour programmer la carte ESP32.
- Traiter les activités telles que :
  - Barrière automatique.
  - Système d'arrosage automatique.

## Recommandations générales

- Avantager les échanges et les discussions autour des solutions proposées.
- Établir des liens et trouver des fils conducteurs entre les différents domaines d'apprentissage rompant ainsi avec l'aspect linéaire.
- Il est préconisé de présenter le contenu à enseigner via des projets, des mini-projets ou des activités utiles ayant un sens pour l'apprenant (jeux, simulation, ...) afin de stimuler l'activité, la collaboration et la créativité chez l'apprenant et favorisant l'aspect interdisciplinaire.
- Il est possible de faire appel à des séquences vidéo, des sites internet, divers documents ou d'une situation réelle, pour dégager l'utilité de l'utilisation des structures de données.
- Il est recommandé de consulter des communautés de développement et de partager des solutions (algorithmes ou programmes) dans des espaces de partage créés pour l'échange et l'apprentissage.
- Favoriser l'exploitation des ressources en ligne.
- Il est important que l'apprenant conserve une trace écrite du travail réalisé en classe. Il appartient à l'enseignant de choisir le support le plus adapté à ses élèves.
- Il est recommandé d'aborder les problèmes et de systématiser leurs résolutions en se basant sur les quatre composantes de la pensée computationnelle : décomposition, reconnaissance de modèles ou de formes, abstraction et algorithme.
- Le langage adopté est Python.

# NIVEAU : 3<sup>EME</sup> ANNEE

## Aide pédagogique 2024-2025

Domaine d'apprentissage	Savoirs associés	Pistes pédagogiques et directives
Pensée Computationnelle et programmation	<ul style="list-style-type: none"><li>• <b>Utiliser les structures algorithmiques adéquates pour résoudre un problème.</b><ul style="list-style-type: none"><li>○ utiliser la structure de données (Les tableaux à une dimension)</li><li>○ utiliser la structure de contrôle conditionnelle à choix multiple (Selon)</li><li>○ utiliser les structures de contrôle répétitives (Tant Que et Répéter).</li></ul></li><li>• <b>Elaborer des solutions algorithmiques modulaires.</b><ul style="list-style-type: none"><li>○ Analyser un problème.</li><li>○ Acquérir la capacité de décomposer un problème en modules.</li><li>○ Identifier les éléments principaux d'un module (paramètres, résultat, type, portée des objets).</li></ul></li></ul>	<ul style="list-style-type: none"><li>• Consolider les notions abordées lors de l'année précédente :<ul style="list-style-type: none"><li>○ Les objets (constantes et variables).</li><li>○ Les types de données (entier, réel, caractère, booléen, chaîne de caractères).</li><li>○ Les structures simples.</li><li>○ Les structures de contrôle (conditionnelles, itérative)</li></ul></li><li>• Exprimer les solutions, selon les besoins, sous forme d'un organigramme, d'une carte mentale, d'un pseudocode, etc.</li><li>• Inciter les apprenants à choisir les structures de données et les structures de contrôle adéquates.</li><li>• Concevoir des solutions algorithmiques.</li><li>• Choisir des exemples concrets pour montrer les avantages de la décomposition modulaire.</li><li>• Argumenter et justifier les choix de la modularité.</li><li>• Inciter les apprenants à écrire des solutions modulaires.</li><li>• Il est conseillé d'habituer les apprenants à commenter les solutions proposées.</li><li>• Utiliser des modules prédéfinis et implémenter des modules personnels.</li><li>• Décrire correctement les entrées / les sorties et le rôle de chaque module.</li></ul>

	<ul style="list-style-type: none"><li>• <b>Utiliser un langage de programmation pour implémenter une solution.</b><ul style="list-style-type: none"><li>○ Analyser un programme existant et lui apporter des modifications pour améliorer sa fonctionnalité.</li><li>○ Implémenter un algorithme en un programme.</li><li>○ Écrire un programme pour résoudre un problème.</li><li>○ Tester, corriger, améliorer et valider un programme.</li></ul></li></ul>	<ul style="list-style-type: none"><li>• Les solutions des problèmes sont implémentées via le langage de programmation Python.</li><li>• L'apprentissage se fait à travers un projet ou des mini projets faisant appel essentiellement aux traitements suivants :<ul style="list-style-type: none"><li>○ Calculs arithmétiques tels que : PGCD, PPCM, nombres premiers, décomposition en facteurs premiers.</li><li>○ Tri d'un tableau avec la méthode de tri à bulles.</li><li>○ Utiliser la méthode de recherche séquentielle pour chercher un élément.</li></ul></li></ul>
--	---	--

## Internet et Nouvelles technologies

### ➤ Exploiter des techniques de développement pour programmer des objets connectés.

- Identifier les concepts de base pour un objet connecté.
- Programmer un objet connecté.

- Faire recours à des supports pédagogiques et didactiques (Vidéos, animations, etc.) pour rappeler les concepts fondamentaux d'un objet connecté.
- Amener l'apprenant à :
  - dégager les composants matériels essentiels d'un objet connecté :
    - les entrées (capteurs, pin, etc.) et les sorties (actionneurs, pin, etc.).
    - l'unité de traitements (ESP32).
    - les supports de stockage (capacité de stockage, mémoire).
    - les moyens de communication (Bluetooth et Wifi).
  - écrire un programme en Micro-Python/Arduino pour échanger des données.
  - téléverser un programme dans la carte ESP32 et tester son fonctionnement.
- Il est recommandé :
  - d'utiliser au moins un capteur (température, humidité, de luminosité, d'obstacle, etc.) pour récupérer des données.
  - d'exploiter les données récupérées pour commander au moins un actionneur (LED, servomoteur, moteur pas à pas, Buzzer, etc.).

## Recommandations générales

- Avantager les échanges et les discussions autour des solutions proposées.
- Établir des liens et trouver des fils conducteurs entre les différents domaines d'apprentissage rompant ainsi avec l'aspect linéaire.
- Il est préconisé de présenter le contenu à enseigner via des projets, des mini-projets ou des activités utiles ayant un sens pour l'apprenant (jeux, simulation, ...) afin de stimuler l'activité, la collaboration et la créativité chez l'apprenant et favorisant l'aspect interdisciplinaire.
- Il est possible de faire appel à des séquences vidéo, des sites internet, divers documents ou d'une situation réelle, pour dégager l'utilité de l'utilisation des structures de données.
- Il est recommandé de consulter des communautés de développement et de partager des solutions (algorithmes ou programmes) dans des espaces de partage créés pour l'échange et l'apprentissage.
- Favoriser l'exploitation des ressources en ligne.
- Il est important que l'apprenant conserve une trace écrite du travail réalisé en classe. Il appartient à l'enseignant de choisir le support le plus adapté à ses élèves.
- Il est recommandé d'aborder les problèmes et de systématiser leurs résolutions en se basant sur les quatre composantes de la pensée computationnelle : décomposition, reconnaissance de modèles ou de formes, abstraction et algorithme.
- Le langage adopté est Python.

# NIVEAU : 4<sup>EME</sup> ANNEE

## Aide pédagogique 2024-2025

Domaine d'apprentissage	Savoirs associés	Pistes pédagogiques et directives
Pensée Computationnelle et programmation	<p><b>Utiliser les structures algorithmiques pour résoudre un problème</b></p> <ul style="list-style-type: none"><li>• <b>Utiliser les structures algorithmiques adéquates pour résoudre un problème.</b><ul style="list-style-type: none"><li>○ Les types de données standards.</li><li>○ Les structures de contrôle conditionnelles.</li><li>○ Les structures de contrôle répétitives.</li><li>○ Les tableaux à une dimension</li></ul></li></ul>	<ul style="list-style-type: none"><li>• Il est important que l'apprenant conserve une trace écrite du travail réalisé en classe.</li><li>• Il est préconisé de présenter le contenu à enseigner via des situations problème et/ou des mini-projets afin de favoriser la collaboration, l'esprit critique et la créativité chez l'apprenant.</li><li>• Il est préconisé de prévoir des situations problèmes afin de résoudre des problèmes faisant appel à des :<ul style="list-style-type: none"><li>○ structures de données : entier, réel, booléen, caractère, chaîne de caractères</li><li>○ structures de contrôles conditionnelles : réduite, complète, Généralisée et à choix multiple</li><li>○ structures itératives : Pour, Répéter et Tant que</li></ul></li><li>• Favoriser l'exploitation des ressources en ligne.</li><li>• Inviter les apprenants à participer à des communautés de développement et de partage de solutions pour une autoformation, pour trouver des réponses à des questionnements ou pour l'enrichir avec leurs productions.</li><li>• Habituer les apprenants à dégager, à partir d'un énoncé, les mots clés permettant de dégager les tâches à réaliser et de déterminer les entrées, les sorties et les pistes des traitements nécessaires pour établir une solution à un problème donné.</li><li>• Dégager les éléments essentiels pour la résolution d'un problème (structures algorithmiques, types de données, traitements, etc.).</li></ul>

	<ul style="list-style-type: none"> <li>● <b>Elaborer des solutions algorithmiques modulaires.</b></li> <li>○ Analyser et modéliser un problème.</li> <li>○ Acquérir la capacité de décomposer un problème en sous-problèmes.</li> <li>○ Identifier les éléments principaux d'un module (entête, paramètres, résultat, type du module, portée des objets, appel).</li> </ul>	<ul style="list-style-type: none"> <li>● Analyser une solution existante et identifier les rôles de différentes structures utilisées.</li> <li>● Inciter les apprenants à expliquer et à argumenter une séquence d'instructions afin de comprendre le traitement.</li> <li>● Inviter les apprenants à identifier, pour un problème donné, une solution parmi plusieurs programmes proposés.</li> <li>● Distinguer les usages et les particularités de chaque type de données.</li> <li>● Il est possible de faire appel à des séquences vidéo, des sites internet, divers documents ou d'une situation réelle pour dégager l'utilité de l'utilisation des structures algorithmiques.</li> <li>● Inviter les apprenants à concevoir des solutions algorithmiques.</li> <li>● Inviter les apprenants à corriger une séquence d'instructions ou une solution erronée.</li> <li>● Avantager les échanges et les discussions autour des solutions proposées.</li> <li>● Apporter les modifications nécessaires à un programme existant pour obtenir un résultat différent.</li> <li>● Inciter les apprenants à identifier et à choisir les structures de données et les structures de contrôle adéquates.</li> <li>● Écrire une solution utilisant les structures conditionnelles.</li> <li>● Écrire une solution utilisant les structures itératives.</li> <li>● Argumenter et justifier les choix des différentes structures.</li> <li>● Habituer les apprenants à commenter les solutions proposées.</li> <li>● Amener l'apprenant à résoudre un problème en le décomposant en modules élémentaires.</li> <li>● Choisir des exemples concrets pour montrer les avantages de la décomposition modulaire.</li> </ul>
--	---	--

		<ul style="list-style-type: none"> <li>• Inciter les apprenants à élaborer des solutions algorithmiques modulaires.</li> <li>• Décrire correctement les entrées / les sorties et le rôle de chaque module.</li> <li>• Argumenter et justifier les choix de la modularité.</li> <li>• Choisir les données appropriées pour tester chaque module.</li> <li>• Prévoir des applications traitant le tri d'un tableau (le tri par sélection et le tri insertion).</li> <li>• Utiliser la méthode de recherche dichotomique pour chercher un élément.</li> <li>• Inscire le développement des programmes dans un travail collaboratif.</li> </ul>
	<p><b>Exploiter un langage de programmation pour implémenter une solution</b></p> <ul style="list-style-type: none"> <li>○ <i>Implémenter un algorithme en un programme.</i></li> <li>○ <i>Écrire un programme pour résoudre un problème.</i></li> <li>▪ <b>Les types de données standards</b></li> <li>▪ <b>Les structures de contrôle.</b></li> <li>▪ <b>Les tableaux à une dimension.</b></li> </ul>	<ul style="list-style-type: none"> <li>• Les solutions des problèmes sont implémentées via le langage de programmation Python.</li> <li>• L'apprentissage est axé principalement sur la pratique.</li> <li>• Écrire un programme en respectant les contraintes de la syntaxe.</li> <li>• Repérer l'erreur de syntaxe dans un programme existant.</li> <li>• Tester, corriger et modifier un programme.</li> <li>• Utiliser ses capacités de raisonnement, ses connaissances sur le langage de programmation, ses savoirs faire et à des outils variés pour améliorer sa solution.</li> <li>• Savoir s'autoévaluer et être capable de décrire ses intérêts, ses compétences et ses acquis.</li> <li>• Amener les apprenants à implémenter les algorithmes solutions des problèmes en Python : <ul style="list-style-type: none"> <li>○ Les structures de données : int, float, bool, str.</li> <li>○ Les structures de contrôles conditionnelle : if, if... else et if... elif... else, match... case.</li> <li>○ Les structures itératives : For, While.</li> </ul> </li> <li>• Ne pas utiliser l'instruction « break » pour la rupture de la boucle.</li> </ul>

	<ul style="list-style-type: none"> <li>○ <i>Tester, corriger, améliorer et valider un programme sur machine.</i></li> <li>○ <i>Utiliser les fonctions prédéfinies de base</i></li> <li>○ <i>Concevoir une interface graphique pour développer des applications simples.</i></li> </ul> <p>▪ <b>Utiliser les objets graphiques les plus usuels (zone texte, label, bouton).</b></p>	<ul style="list-style-type: none"> <li>● Habituer les apprenants à appliquer les bonnes pratiques de programmation (nomenclature des objets, indentation de la solution, commentaire, etc.).</li> <li>● Lors de l'implémentation, l'apprenant devrait se conformer aux recommandations de l'enseignant quant à l'usage des méthodes, étapes, formules.</li> <li>● Utiliser le langage de programmation Python pour développer des interfaces graphiques pour des applications simples.</li> <li>● Installer le logiciel Qt designer et les bibliothèques nécessaires.</li> <li>● Découvrir l'interface graphique à partir d'une application existante.</li> <li>● Concevoir et utiliser une interface utilisateur graphique (GUI)</li> <li>● Initier les apprenants au principe de la programmation événementielle.</li> <li>● Utiliser la technique « Glisser-Déposer » pour la conception d'une interface graphique et la programmation des objets se fait à l'aide du langage python.</li> <li>● L'usage de l'interface graphique doit être fait dans le but d'améliorer l'ergonomie de quelques programmes, mais en aucun cas un objet principal d'apprentissage.</li> <li>● Traiter des problèmes faisant appel à une seule interface.</li> <li>● Amener l'apprenant à identifier les objets graphiques à utiliser.</li> <li>● Se limiter aux méthodes suivantes pour les objets graphiques : <ul style="list-style-type: none"> <li>○ .text, .setText, .clear, .show, .clicked</li> </ul> </li> </ul>
--	--	---

## Recommandations générales

- Avantager les échanges et les discussions autour des solutions proposées.
- Il est possible de faire appel à des séquences vidéo, des sites internet, divers documents ou d'une situation réelle, pour dégager l'utilité de l'utilisation des structures de données.
- Il est judicieux d'utiliser la pédagogie active et de traiter divers problèmes de la vie courante (mathématiques, physiques, économies ...)
- Il est recommandé d'aborder les problèmes et de systématiser leurs résolutions en se basant sur les quatre composantes de la pensée computationnelle : décomposition, reconnaissance de modèles ou de formes, abstraction et algorithme.
- Il est recommandé de consulter des communautés de développement et de partager des solutions (algorithmes ou programmes) dans des espaces de partage créés pour l'échange et l'apprentissage.
- Favoriser l'exploitation des ressources en ligne.
- Il est important que l'apprenant conserve une trace écrite du travail réalisé en classe. Il appartient à l'enseignant de choisir le support le plus adapté à ses élèves.
- Le langage adopté est Python.