

République Tunisienne
Ministère de l'Education



Aides pédagogiques **D'INFORMATIQUE** **SPECIFIQUES**

**SECTIONS : SCIENCES EXPERIMENTALES,
SCIENCES TECHNIQUES
ET MATHEMATIQUES**

Septembre 2022

NIVEAU : 2^{EME} ANNEE

Aide pédagogique 2022-2023

Domaine d'apprentissage	Savoirs associés	Pistes pédagogiques et directives
<p>Pensée Computationnelle et programmation</p>	<ul style="list-style-type: none">● Identifier les étapes de résolution d'un problème.<ul style="list-style-type: none">○ Dégager les éléments essentiels permettant la résolution d'un problème.● Élaborer des solutions sous forme d'algorithmes.<ul style="list-style-type: none">○ Utiliser des structures de données à bon escient.○ Utiliser les structures de contrôle adéquates pour résoudre un problème.○ Traiter essentiellement les notions suivantes :<ul style="list-style-type: none">▪ Les objets (constantes et variables)▪ Les types de données (entier, réel, caractère, booléen, chaîne).▪ Les structures simples.	<ul style="list-style-type: none">● Etablir des liens et trouver des fils conducteurs entre les différents domaines d'apprentissage rompant ainsi avec l'aspect linéaire du programme.● Il est préconisé de présenter les savoirs associés à travers des projets, des mini-projets ou des activités ayant du sens pour l'apprenant.● Favoriser l'investigation, le questionnement, l'apprentissage expérientiel, l'apprentissage par problème. etc.● 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.● Donner du sens aux activités, les diversifier et opter pour une démarche interdisciplinaire permettant le décroisement entre les divers champs d'apprentissages et l'ouverture de l'informatique sur les autres disciplines.● Favoriser l'exploitation des ressources en ligne et développer la communication.● Inciter à l'innovation et motiver les apprenants pour la créativité.● Inciter les apprenants à faire des échanges autour des solutions proposées et de les partager en ligne.● Il est pertinent de :<ul style="list-style-type: none">○ familiariser l'apprenant à formuler sous forme d'actions des solutions à des problèmes puisés de son vécu. On pourra exprimer ces solutions à

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

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**
 - Connaitre les interfaces de l'objet à connecter.
 - Savoir 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.
- Traiter les activités telles que :
 - Faire clignoter une diode Led.
 - Feu de carrefour (rouge, vert, orangé).

NIVEAU : 3^{EME} ANNEE

Aide pédagogique 2022-2023

Domaine d'apprentissage	Savoirs associés	Pistes pédagogiques et directives
<p style="text-align: center;">Pensée Computationnelle et programmation</p>	<ul style="list-style-type: none"> • Utiliser les structures algorithmiques adéquates pour résoudre un problème. <ul style="list-style-type: none"> ○ Les objets (constantes et variables). ○ Les types de données (entier, réel, caractère, booléen, chaîne de caractères). ○ Les structures simples. ○ Les structures de contrôle (Si, Selon, Pour, Tant Que et Répéter). ○ Les tableaux à une dimension. • Elaborer des solutions algorithmiques modulaires. <ul style="list-style-type: none"> ○ Analyser un problème. ○ Acquérir la capacité de décomposer un problème en modules. ○ Identifier les éléments principaux d'un module (paramètres, résultat, type, portée des objets). 	<ul style="list-style-type: none"> • Exprimer les solutions, selon les besoins, sous forme d'un organigramme, d'une carte mentale, d'un pseudocode, etc. • Inciter les apprenants à choisir les structures de données et les structures de contrôle adéquates. • Concevoir des solutions algorithmiques. • Choisir des exemples concrets pour montrer les avantages de la décomposition modulaire. • Argumenter et justifier les choix de la modularité. • Inciter les apprenants à écrire des solutions modulaires. • Il est conseillé d'habituer les apprenants à commenter les solutions proposées. • Utiliser des modules prédéfinis et implémenter des modules personnels. • Décrire correctement les entrées / les sorties et le rôle de chaque module.
	<ul style="list-style-type: none"> • Utiliser un langage de programmation pour implémenter une solution. <ul style="list-style-type: none"> ○ Analyser un programme existant et lui apporter des modifications pour améliorer sa fonctionnalité. ○ Implémenter un algorithme en un programme. 	<ul style="list-style-type: none"> • Les solutions des problèmes sont implémentées via le langage de programmation Python. • L'apprentissage se fait à travers un projet ou des mini projets faisant appel essentiellement aux traitements suivants :

	<ul style="list-style-type: none"> ○ Écrire un programme pour résoudre un problème. ○ Tester, corriger, améliorer et valider un programme. 	<ul style="list-style-type: none"> ○ Calculs arithmétiques tels que : PGCD, PPCM, nombres premiers, décomposition en facteurs premiers. ○ Tri d'un tableau (la méthode de tri à bulles). ○ Recherche d'un élément dans un tableau.
<p>Internet et Nouvelles technologies</p>	<ul style="list-style-type: none"> ● Prendre conscience de l'intérêt de la robotique <ul style="list-style-type: none"> - Définir la robotique. - Identifier des domaines d'application de la robotique. ● Piloter un objet connecté <ul style="list-style-type: none"> - Connaître les interfaces de l'objet à connecter. - Savoir connecter un objet à l'ordinateur - Programmer des objets simples virtuels ou réels pour réaliser différentes tâches d'une façon innovante. 	<ul style="list-style-type: none"> ● 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. ● Traiter les activités telles que : <ul style="list-style-type: none"> ○ Faire clignoter une diode Led. ○ Feu de carrefour (rouge, vert, orangé).

NIVEAU : 4^{EME} ANNEE

Aide pédagogique 2022-2023

Domaine d'apprentissage	Savoirs associés	Pistes pédagogiques et directives
<p>Pensée Computationnelle et programmation</p>	<p>Utiliser les structures algorithmiques pour résoudre un problème</p> <ul style="list-style-type: none"> • <i>Utiliser les structures algorithmiques adéquates pour résoudre un problème.</i> <ul style="list-style-type: none"> ○ Les types de données standards. ○ Les structures de contrôle conditionnelles. ○ Les structures de contrôle répétitives. ○ Les tableaux à une dimension • Elaborer des solutions algorithmiques modulaires. 	<ul style="list-style-type: none"> • Il est important que l'apprenant conserve une trace écrite du travail réalisé en classe. • 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. • Il est préconisé de prévoir des situations problèmes afin de résoudre des problèmes faisant appel à des : <ul style="list-style-type: none"> ○ structures de données : entier, réel, booléen, caractère, chaîne de caractères ○ structures de contrôles conditionnelles : réduite, complète, Généralisée et à choix multiple ○ structures itératives : Pour, Répéter et Tant que • Favoriser l'exploitation des ressources en ligne. • 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. • 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. • Dégager les éléments essentiels pour la résolution d'un problème (structures algorithmiques, types de données, traitements, etc.).

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

		<ul style="list-style-type: none"> • Choisir les données appropriées pour tester chaque module. • Prévoir des applications traitant le tri d'un tableau (le tri par sélection et le tri insertion). • Inscrire le développement des programmes dans un travail collaboratif.
	<p>Exploiter un langage de programmation pour implémenter une solution</p> <ul style="list-style-type: none"> ○ <i>Implémenter un algorithme en un programme.</i> ○ <i>Écrire un programme pour résoudre un problème.</i> <ul style="list-style-type: none"> ▪ Les types de données standards <ul style="list-style-type: none"> ▪ Les structures de contrôle. ▪ Les tableaux à une dimension. ○ <i>Tester, corriger, améliorer et valider un programme sur machine.</i> ○ <i>Utiliser les fonctions prédéfinies de base</i> 	<ul style="list-style-type: none"> • Les solutions des problèmes sont implémentées via le langage de programmation Python. • L'apprentissage est axé principalement sur la pratique. • Écrire un programme en respectant les contraintes de la syntaxe. • Repérer l'erreur de syntaxe dans un programme existant. • Tester, corriger et modifier un programme. • Utiliser ses capacités de raisonnement, ses connaissances sur le langage de programmation, ses savoirs faire et à des outils variés pour améliorer sa solution. • Savoir s'autoévaluer et être capable de décrire ses intérêts, ses compétences et ses acquis. • Amener les apprenants à implémenter les algorithmes solutions des problèmes en Python : <ul style="list-style-type: none"> ○ Les structures de données : int, float, bool, Str. ○ Les structures de contrôles conditionnelle : if, if... else et if... elif... else, match... case. ○ Les structures itératives : For, While. • Ne pas utiliser l'instruction « break » pour la rupture de la boucle. • Habituer les apprenants à appliquer les bonnes pratiques de programmation (nomenclature des objets, indentation de la solution, commentaire, etc.). • Lors de l'implémentation, l'apprenant devrait se conformer aux recommandations de l'enseignant quant à l'usage des méthodes, étapes, formules. • Utiliser le langage de programmation Python pour développer des interfaces graphiques pour des applications simples. • Installer le logiciel Qt designer et les bibliothèques nécessaires.

	<ul style="list-style-type: none">○ <i>Concevoir une interface graphique pour développer des applications simples.</i><ul style="list-style-type: none">▪ Utiliser les objets graphiques les plus usuels (zone texte, label, bouton).	<ul style="list-style-type: none">• Découvrir l'interface graphique partir d'une application existante.• Concevoir et utiliser une interface utilisateur graphique (GUI)• Initier les apprenants au principe de la programmation événementielle.• 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.• 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.• Traiter des problèmes faisant appel à une seule interface.• Amener l'apprenant à identifier les objets graphiques à utiliser.• Se limiter aux méthodes suivantes pour les objets graphiques :<ul style="list-style-type: none">○ .text, .setText, .clear, .show, .clicked
--	--	---