

République Tunisienne
Ministère de l'Éducation



Aides pédagogiques **D'INFORMATIQUE**

SECTION : SCIENCES DE L'INFORMATIQUE

Septembre 2024

NIVEAU : 2^{EME} ANNEE

Matière : Informatique

Aide pédagogique 2024-2025

Domaines d'apprentissage	Compétences et savoirs associés détaillés	Pistes pédagogiques et directives
Pensée computationnelle et programmation	<ul style="list-style-type: none"> • Reconnaître les phases de résolution d'un problème. <ul style="list-style-type: none"> - Lire et comprendre l'énoncé d'un problème afin de dégager les tâches à réaliser. - Dégager les éléments essentiels pour la résolution (les entrées, les sorties et les traitements). - Élaborer une solution sous forme d'un algorithme. - Écrire et exécuter le programme solution sur ordinateur. - Apporter des modifications à la solution (actions correctives, actions évolutives). • Décomposer un problème en modules. <ul style="list-style-type: none"> - Identifier des sous-problèmes pertinents (modules). - Identifier les éléments principaux d'un module (Type, paramètres, résultat, etc.). - Acquérir la capacité de décomposer un problème en sous-problèmes : décomposition logique. • Exploiter des concepts algorithmiques pour résoudre des problèmes. <ul style="list-style-type: none"> - Utiliser des structures de données à bon escient : • Dégager les objets nécessaires (variables/constantes) pour résoudre un problème. • Distinguer les usages et les particularités de chaque type de données, afin 	<ul style="list-style-type: none"> • Il est possible de faire appel à des séquences vidéo, des sites internet, divers documents ou d'une situation réelle, ... pour dégager les phases de résolution d'un problème. • Il est recommandé d'utiliser des exemples concrets pour montrer les avantages de la décomposition (meilleure lisibilité, diminution de risque d'erreurs, réutilisation de modules dans un ou plusieurs algorithmes, simplicité de l'entretien, favorisation de travail en équipe). • Chaque niveau de décomposition est suivi par l'élicitation (valorisation, argumentation, justification) de sous problèmes. • L'initiation à l'algorithmique peut se faire à partir d'un algorithme existant (structure d'un algorithme et tournage à

d'utiliser le plus adapté pour déclarer un objet nécessaire dans la résolution d'un problème donné (Entier, Réel, Booléen, Caractère, Chaîne de caractères et Tableau à une dimension).

- Utiliser les structures simples pour lire des données, pour afficher des informations et pour attribuer une valeur à une variable.
- Utiliser les structures de contrôle adéquates pour résoudre un problème.
 - Utiliser les structures conditionnelles pour effectuer des choix en fonction des circonstances.
 - Utiliser les structures répétitives pour répéter un ensemble d'instructions autant de fois que nécessaire.
- Utiliser un langage de programmation pour implémenter une solution.
 - Traduire un algorithme en un programme exécutable.
 - Écrire un programme pour résoudre un problème.
- Tester une solution implémentée afin de répondre à un besoin spécifique.
 - Exécuter une solution implémentée.
 - Modifier un code de programmation existant pour changer le comportement d'un programme.

la main).

- Il est nécessaire d'habituer les apprenants à exploiter à bon escient les structures de données (Objets et types) et les structures de contrôle lors de la résolution d'un problème (nombre de variables, nombre d'instructions, structure de contrôle adéquate, etc.)
- il est nécessaire de se limiter aux traitements adaptés au niveau des apprenants.
- On pourra utiliser des outils d'exécution d'algorithmes tels que "Algobox", "Larp", etc.
- Inciter les apprenants à comparer différents algorithmes pouvant résoudre le même problème.
- Toutes les solutions des problèmes sont implémentées via le langage de programmation **Python**.
- On pourra utiliser des outils tels que Trinket.io et Pencilcode.net
- L'initiation à l'utilisation du langage peut se faire à partir d'un programme existant (structure d'un programme, exécution et exploration du code).
- Il est possible de traduire un algorithme existant en un programme.

		<ul style="list-style-type: none"> • Il est utile d'inciter les apprenants à analyser un programme exécutable afin de comprendre les traitements. • Se servir de dispositifs ou de robots pour appliquer des notions de programmation, en mettant à profit différents outils et langages de programmation. • Ecrire un programme en Micro-Python ou Arduino pour programmer une carte Esp32 afin de réaliser différentes tâches. • Il est essentiel d'habituer les apprenants à commenter les solutions.
<p align="center">Systemes, technologies et Internet</p>	<ul style="list-style-type: none"> • Exploiter des techniques de développement pour créer des documents web. <ul style="list-style-type: none"> - Créer un document web <ul style="list-style-type: none"> ▪ S'appropriier le vocabulaire et la syntaxe du langage HTML5 en créant des pages web significatives. <ul style="list-style-type: none"> ✓ Identifier la structure de base d'une page web : <code><!doctype></code>, <code><html></code>, <code><head></code> et <code><body></code> . ✓ Manipuler les éléments du langage Html5. <ul style="list-style-type: none"> ○ Organiser un texte de manière sémantique : <code><p></code>, <code>
</code>, <code><h_n></code>, <code><mark></code>, <code></code>, <code></code>, <code></code> <code><table></code>, <code><tr></code>, <code><td></code>, <code><th></code>, <code><tfoot></code>, <code><thead></code>, <code><tbody></code>. ○ Intégrer des éléments multimédias : <code></code>, <code><audio></code>, <code><video></code>. ○ Intégrer des liens : <code><a></code> . 	<ul style="list-style-type: none"> • Exploiter un éditeur Web WYSIWYG qui intègre le HTML5. • Découvrir la structure de base d'un document HTML5 en explorant des pages web existantes. • Habituer les apprenants à indenter et formater correctement le code et à le commenter.

	<ul style="list-style-type: none"> ○ Créer/Modifier un formulaire en utilisant les éléments : <form>, <input> de type (text, number, radio, checkbox, reset, submit), <select>, <option>, <fieldset>, <legend>. ▪ Exploiter les techniques appropriées pour appliquer des mises en forme à une page web. <ul style="list-style-type: none"> ✓ Reconnaître les propriétés de mises en forme (texte, bordure et arrière plan). ✓ Appliquer des styles : <ul style="list-style-type: none"> ○ en ligne (inline) sur des éléments d'une page web. ○ internes (head) sur des éléments d'une page web. <ul style="list-style-type: none"> • Reconnaître la syntaxe d'une règle CSS3 (sélecteur, propriété et valeur). • Déclarer une règle CSS3 sur des éléments HTML. - Valider le contenu HTML5. <ul style="list-style-type: none"> ▪ Utiliser des outils de validation du contenu des pages web. Corriger les erreurs et les avertissements détectés par les validateurs. 	<ul style="list-style-type: none"> • Pour chaque élément html, traiter les attributs qui lui correspondent (voir annexe). • Prévoir des activités basées sur la variation de la valeur de l'attribut style pour mettre en forme des éléments d'une page web créée. • Inciter les apprenants à découvrir la syntaxe d'une règle CSS3 (déclaration, sélecteur, propriété et valeur) en explorant des pages web existantes.
--	--	---

Recommandations générales

- Avantager les échanges et les discussions autour des solutions proposées.
- Etablir des liens et trouver des fils conducteurs entre les différents domaines d'apprentissage rompant ainsi avec l'aspect linéaire de sa mise en œuvre.
- Il est préconisé de présenter le contenu à enseigner via des projets, des mini-projets ou des activités, ayant un sens pour l'apprenant (jeux, simulation, ...) et stimulant chez lui l'activité, la collaboration et la créativité ; tout en favorisant l'aspect interdisciplinaire.
- 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.

NIVEAU : 3^{EME} ANNEE
Matière : Algorithmique & programmation
Aide pédagogique 2024-2025

Domaine d'apprentissage	Compétences et savoirs associés détaillés	Pistes pédagogiques et directives
Pensée Computationnelle et programmation	<ul style="list-style-type: none"> • Exploiter des concepts algorithmiques avancés pour résoudre des problèmes. - Lire et comprendre l'énoncé d'un problème afin de dégager les tâches à réaliser. - Dégager les éléments essentiels pour la résolution d'un problème (structures et types de données, traitements). - Distinguer les usages et les particularités de chaque type de données, afin d'utiliser le plus adapté pour déclarer un objet nécessaire dans la résolution d'un problème donné. - Utiliser des structures de données avancées pour résoudre un problème (Tableau à deux dimensions, Enregistrement). 	<ul style="list-style-type: none"> - 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 avancées. - Il est nécessaire d'habituer les apprenants à : <ul style="list-style-type: none"> ○ exploiter à bon escient les structures de données avancées (Objets et types). ○ appliquer les bonnes pratiques de programmation (nomenclature des objets, commentaire, etc.). - Traiter la méthode de tri par sélection et la méthode de tri à bulles - Traiter la recherche d'un élément (séquentielle et dichotomique).

	<ul style="list-style-type: none"> - Apporter les modifications nécessaires à une solution pour répondre à un besoin. - Exploiter des concepts algorithmiques avancés pour résoudre des problèmes : <ul style="list-style-type: none"> ○ d'arithmétiques ○ d'optimisation et d'approximation ● Utiliser un environnement de programmation pour implémenter une solution. <ul style="list-style-type: none"> - Implémenter un algorithme en un programme exécutable. - Écrire un programme pour résoudre un problème. ● Concevoir une interface graphique pour développer des applications simples comportant les objets suivants : fenêtre, zone de texte (Text Edit, Line Edit), bouton (Push Button), bouton radio (Radio Button), liste déroulante (Combo Box), case à cocher (Check Box), étiquette (label). 	<ul style="list-style-type: none"> - Traiter essentiellement : <ul style="list-style-type: none"> ○ Des calculs arithmétiques (PGCD, PPCM, nombres premiers, décomposition en facteurs premiers, etc.) ○ Des problèmes d'optimisations ○ Des méthodes de calcul d'une valeur approchée de constantes connues (π, e, ...) - Le langage adopté est Python. - Tester le programme solution sur ordinateur. - Apporter des modifications à une solution (actions correctives, actions évolutives). - Il est utile d'inciter les apprenants à analyser un programme exécutable afin de comprendre ses traitements. - La découverte d'une interface graphique peut se faire à partir d'une application existante. - La conception d'une interface graphique se fait en utilisant la technique « Glisser-Déposer » (Drag & drop). - Utiliser Qtdesigner comme outil de création d'interfaces graphiques. - Tenir compte des contrôles de saisie des champs d'une interface graphique en affichant des messages d'erreurs. - L'apprentissage est axé principalement sur la pratique.
--	--	---

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 avancées.
- 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^{EME} ANNEE
Matière : Algorithmique & programmation
Aide pédagogique 2024-2025

Domaines d'apprentissage	Compétences et savoirs associés détaillés	Pistes pédagogiques et directives
Pensée computationnelle et programmation	<ul style="list-style-type: none"> • Exploiter des concepts algorithmiques avancés pour résoudre des problèmes faisant appel à : <ul style="list-style-type: none"> ○ des structures de données <ul style="list-style-type: none"> ▪ Fichier texte ▪ Fichier typé ○ des méthodes de tri <ul style="list-style-type: none"> ▪ Tri par insertion ▪ Tri shell ○ la récursivité 	<ul style="list-style-type: none"> - Rappeler les notions tableau à deux dimensions et enregistrement à travers la résolution de problème exploitant ces structures de données. - Traiter les notions fichier texte et fichier typé à travers la résolution de problème exploitant ces structures de données. - Utiliser la mémoire centrale pour effectuer les traitements modifiant le contenu initial d'un fichier tels que le tri, l'insertion d'un élément, la suppression d'un élément, le décalage, ... - Rappeler les deux méthodes de tri : Le tri par sélection et le tri à bulles. - On pourra traiter d'autres algorithmes de tri (tri par création, tri par comptage, ...). - Montrer, quand c'est possible, le passage d'une formulation itérative à une formulation récursive. - Ne traiter que le cas de récursivité simple (ni croisée, ni indirecte) sur des problèmes naturellement récursifs (factorielle, palindrome, PGCD, ...)

	<ul style="list-style-type: none"> ○ des traitements récurrents et arithmétiques ○ l'optimisation et l'approximation ● Utiliser un environnement de programmation pour implémenter une solution. <ul style="list-style-type: none"> ○ Implémenter un algorithme en un programme exécutable. ○ Écrire un programme pour résoudre un problème. 	<ul style="list-style-type: none"> - On traitera divers problèmes en axant sur la relation de récurrence d'ordre un et plus (suites, triangle de pascal, le nombre d'or, ...) - Traiter essentiellement : <ul style="list-style-type: none"> ○ la suite de Fibonacci ○ calcul du factoriel, PGCD, PPCM, nombre premier, décomposition en facteurs premiers. ○ les conversions entre bases de numération ○ les calculs de $C(n,p)$ et de $A(n,p)$ - Traiter essentiellement : <ul style="list-style-type: none"> ○ des problèmes d'optimisations ○ la recherche du zéro d'une fonction ($f(x)=0$) ○ la recherche du point fixe d'une fonction ($f(x)=x$) ○ des méthodes de calcul d'une valeur approchée de constantes connues (π, e, \dots) ○ calcul d'aires (rectangles, trapèzes) - Le langage adopté est Python. - Il est utile d'inciter les apprenants à analyser un programme exécutable afin de comprendre les traitements. - Il est essentiel d'habituer les apprenants à commenter les solutions. - Apporter des modifications à une solution (actions correctives, actions évolutives).
--	--	--

	<ul style="list-style-type: none">○ Concevoir une interface graphique pour développer des applications simples comportant les objets suivants :<ul style="list-style-type: none">▪ List Widget▪ Table Widget	<ul style="list-style-type: none">- Tester le programme solution sur ordinateur.- Rappeler les composants d'une interface graphique à partir d'une application existante (fenêtre, Label, liste déroulante (Combo Box), Zone de texte (Text Edit, Line Edit), bouton (Push Button), bouton radio (Radio Button), Case à cocher (Check Box)).- La conception d'une interface graphique se fait en utilisant la technique « Glisser-Déposer » (Drag & drop).- Utiliser Qt designer comme outil de création d'interfaces graphiques.- Tenir compte des contrôles de saisie des champs d'une interface graphique en affichant des messages d'erreurs.- Il est recommandé d'utiliser des fichiers pour transférer et récupérer des informations.- L'apprentissage est axé principalement sur la pratique.
--	---	--

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 avancées.
- 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.

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 have a subtle pattern of small white dots. The squares are arranged in a cluster that tapers towards the top right, creating a sense of depth and movement.

Aides pédagogiques **D'INFORMATIQUE**

SECTION : SCIENCES DE L'INFORMATIQUE

Septembre 2024

NIVEAU : 3^{EME} ANNEE

Matière : Systèmes & Technologies de l'Informatique

Aide pédagogique 2024-2025

Domaine d'apprentissage	Compétences et savoirs associés	Pistes pédagogiques et directives
Systèmes, technologies et Internet	<p>➤ Exploiter des techniques de développement pour programmer des objets connectés.</p> <ul style="list-style-type: none">• Programmer un objet connecté.<ul style="list-style-type: none">- Écrire un programme pour interagir avec le matériel.<ul style="list-style-type: none">✓ Configurer la communication réseau de l'objet connecté.✓ Echanger des données via l'objet connecté.- Tester le programme de l'objet connecté et le corriger.	<ul style="list-style-type: none">• Réviser et consolider les notions clés abordées les années précédentes via des activités/exercices issues de la vie courante de l'apprenant en relation avec :<ul style="list-style-type: none">- les concepts fondamentaux d'un objet connecté.- les composants matériels essentiels d'un objet connecté : les entrées, les sorties , l'unité de traitement, les supports stockage, les moyens de communication.- l'écriture d'un programme en Micro-Python ou Arduino pour échanger des données au sein d'un réseau Personnel/Local.- Le téléversement d'un programme dans la carte ESP32 pour tester son fonctionnement.• Il est recommandé :<ul style="list-style-type: none">- d'utiliser essentiellement les capteurs de température/humidité, de luminosité et d'obstacle pour récupérer des données.- d'exploiter les données récupérées pour commander essentiellement les actionneurs suivants : LED, servomoteur, moteur pas à pas et Buzzer.

	<p>➤ Exploiter des fonctionnalités d'un système d'exploitation pour gérer l'environnement informatique de travail.</p> <ul style="list-style-type: none"> • Identifier différents types de systèmes d'exploitation. <ul style="list-style-type: none"> - Définir un système d'exploitation. - Distinguer des types de systèmes d'exploitation (PC, Mobile et embarqué). 	<ul style="list-style-type: none"> • Faire recours à des supports pédagogiques et didactiques (Vidéos, animations, etc..) pour amener les apprenants à dégager la définition d'un système d'exploitation et reconnaître les types suivants : <ul style="list-style-type: none"> - systèmes d'exploitation pour PC (Windows, Linux et MAC OS). - systèmes d'exploitation pour Mobiles (Android et IOS). - systèmes d'exploitation embarqués (WatchOS, TVOS, QNX, etc.).
	<ul style="list-style-type: none"> • Manipuler des fonctionnalités d'un système d'exploitation pour un ordinateur : <ul style="list-style-type: none"> - Déployer un système d'exploitation (Installer, configurer et mettre à jour). - Sécuriser un ordinateur en utilisant des logiciels de protection. 	<ul style="list-style-type: none"> • Impliquer les apprenants dans l'utilisation d'une machine virtuelle (VMWare, ...) pour manipuler des systèmes d'exploitation pour PC (Propriétaire et Libre). • Amener l'apprenant à sécuriser un système d'exploitation en exploitant des logiciels de protection (Antivirus, etc.).
	<p>➤ Acquérir des notions de base d'un réseau pour assurer la communication d'un objet dans un réseau local.</p> <ul style="list-style-type: none"> • Définir un réseau. • Classifier des réseaux selon leur étendu (LAN, WAN et MAN). • Paramétrer un composant d'un réseau (adresse IP, serveurs DNS, masque réseau et adresse MAC). 	<ul style="list-style-type: none"> • Amener l'apprenant à reconnaître les architectures des réseaux client/serveur et poste à poste. • sécuriser un réseau en configurant le Pare-feu et la liste d'accès pour sécuriser les postes de travail d'un réseau.

	<p>➤ Exploiter des techniques de développement pour créer et publier un site web interactif.</p> <ul style="list-style-type: none"> • Créer un site web. <ul style="list-style-type: none"> - Créer des documents web en utilisant les éléments HTML5. <ul style="list-style-type: none"> ○ Utiliser les balises d'en-tête pour : <ul style="list-style-type: none"> ✓ Lier des feuilles de style CSS(<link>). ✓ Ajouter des scripts JavaScript (<script>). ○ Manipuler les conteneurs HTML : <ul style="list-style-type: none"> ✓ Structurer le contenu : <div>, . ✓ Intégrer du contenu externe <iframe>. ○ Structurer une page web sémantiquement : <ul style="list-style-type: none"> ✓ Organiser le contenu de manière sémantique à l'aide de : <main>, <article>, <section>, <nav>, <header>, <footer> et <aside>. ✓ Gérer le texte sémantiquement à l'aide de : <address>, <cite> et <output>. ✓ Ajouter de l'interactivité et organiser des informations supplémentaires à l'aide de : <details> et <summary>. ○ Intégrer des éléments multimédias à l'aide de : <figure>, <figcaption> et <source>. ○ Créer/Modifier des formulaires en intégrant les éléments d'entrée : date, time, email, tel, password, range et button. ○ Utiliser les événements : onfocus, onblur, onclick, oninput, onchange et onload. 	<ul style="list-style-type: none"> • Exploiter un éditeur Web WYSIWYG qui intègre le HTML5. • Réviser et consolider les notions clés abordées l'année précédente via des activités/exercices issues de la vie courante de l'apprenant en relation avec : <ul style="list-style-type: none"> - L'organisation d'un texte de manière sémantique. - L'intégration des éléments multimédia. - L'intégration des liens - La création des formulaires. - La validation d'un document HTML5. - La syntaxe d'une règle CSS3. - Les propriétés de mises en forme (texte, bordure et arrière-plan). • Pour chaque élément html, traiter les attributs qui lui correspondent (voir annexe).
	<ul style="list-style-type: none"> - Appliquer des styles CSS3 pour mettre en forme une page web. <ul style="list-style-type: none"> ○ Utiliser des feuilles de style. ○ Déclarer des sélecteurs CSS : Sélecteurs d'élément (de type), d'identifiant (#id), de classe (.class), de groupe et universel (*). 	

	<ul style="list-style-type: none"> ○ Mettre en forme du texte. ○ Styliser l'arrière-plan. ○ Gérer les bordures d'un élément. ○ Positionner et dimensionner un élément. ○ Appliquer des animations et des transitions. <ul style="list-style-type: none"> ✓ Créer des <i>animations CSS3</i>. ✓ Ajouter un effet de transition à un élément. 	
	<ul style="list-style-type: none"> - Utiliser le langage JavaScript pour ajouter de l'interactivité dans un site web. <ul style="list-style-type: none"> ○ Repérer l'emplacement d'un script JavaScript dans une page web. ○ Intégrer un script JavaScript dans une page à partir d'un fichier externe. ○ Manipuler les actions élémentaires simples : Entrées (prompt), sorties (innerHTML, write, alert) et affectation. ○ Accéder à un élément d'un formulaire via son identifiant (getElementById). ○ Manipuler des variables et des constantes : <ul style="list-style-type: none"> ✓ Déclarer une constante (const) et une variable d'une manière implicite et/ou explicite (let). ✓ Utiliser les types de données : string, number, boolean, array et date. ✓ Distinguer la portée des variables (locale et globale). ○ Utiliser des structures de contrôle : <ul style="list-style-type: none"> ✓ Conditionnelles : if, else, else if et switch. ✓ Itératives : for, while et do...while. ○ Manipuler des modules. <ul style="list-style-type: none"> ✓ Déclarer un module. ✓ Appeler un module 	<ul style="list-style-type: none"> • Amener les apprenants à identifier l'apport de l'intégration du langage JavaScript dans une page web. • Il est possible de découvrir la syntaxe JavaScript à partir de pages existantes ou de l'éditeur utilisé. • Implémenter le code JavaScript dans une page web et dans un fichier externe. • Prévoir des traitements sur les objets Date, String, Array, Number et Math. • Amener les apprenants à identifier les structures de données et de contrôles appropriées répondant aux besoins spécifiques du site web. • Il est recommandé de présenter les traitements sur les champs du formulaire sous forme de modules.

	<ul style="list-style-type: none"> - Assurer la validité du code HTML5 et CSS3. <ul style="list-style-type: none"> o Utiliser des outils de validation. o Corriger les erreurs et les avertissements détectés par les validateurs. 	
<p style="text-align: center;">Gestion de données</p>	<ul style="list-style-type: none"> ➤ Créer et gérer des bases de données. <ul style="list-style-type: none"> • Identifier les concepts fondamentaux d'une base de données relationnelle (BDR). <ul style="list-style-type: none"> - Définir une BDR. - Reconnaître l'utilité d'une BDR. - Définir un Système de Gestion d'une BDR. - Reconnaître les fonctionnalités d'un SGBDR. - Définir les notions : table, relation (un à un ; un à plusieurs ; plusieurs à plusieurs), enregistrement, champ, clé primaire, clé étrangère et contraintes d'intégrité. - Distinguer les contraintes d'intégrités (de table, de domaine et référentielle). - Identifier les propriétés d'un champ (nom, type, taille et contraintes). • Manipuler la structure d'une BDR. <ul style="list-style-type: none"> - Créer une BDR. - Ajouter, modifier et supprimer des tables/colonnes/contraintes d'intégrité. • Manipuler les données d'une BDR (consultation, ajout, suppression et modification). 	<ul style="list-style-type: none"> • Recourir à des exemples de BDR simplifiées et composées au maximum de quatre tables et touchant le vécu de l'apprenant. • Gérer une BDR en utilisant le SGBDR MySQL. • Se référer au schéma relationnel, la description des colonnes et les données pour créer et manipuler une BDR. • Solliciter les apprenants à analyser les informations qui présentent une BDR. • Adopter, pour la représentation textuelle et graphique d'une BDR, la convention suivante : <ul style="list-style-type: none"> - clé primaire soulignée, - clé étrangère suivie du symbole #. • Utiliser la notation suivante pour présenter une relation : <ul style="list-style-type: none"> - un à un présentée par 1 : 1. - un à plusieurs présentée par 1 : ∞. - plusieurs à plusieurs présentée par ∞ : ∞. • Inviter les apprenants à appliquer des contraintes d'intégrités. • Inciter les apprenants à découvrir toute commande SQL générée par l'outil visuel lors de la manipulation de la BDR. • Engager les apprenants à élaborer des requêtes SQL simples (sans faire recours aux jointures et l'utilisation des fonctions agrégats).

NIVEAU : 4^{EME} ANNEE

Matière : Systèmes & Technologies de l'Informatique

Aide pédagogique 2024-2025

Domaine d'apprentissage	Compétences et savoirs associés	Pistes pédagogiques et directives
Gestion de données	<p>➤ Acquérir des concepts fondamentaux de base de données pour gérer les données.</p> <ul style="list-style-type: none">• S'approprier la terminologie et les concepts fondamentaux d'une BD relationnelle (BDR).<ul style="list-style-type: none">- Appliquer des contraintes d'intégrité dans une BDR (contrainte de table, de domaine et d'intégrité référentielle).- Convertir une représentation graphique d'une base de données en une représentation textuelle et inversement.- Evaluer une représentation (textuelle/graphique) d'une BDR.- Corriger une représentation (textuelle/graphique) d'une BDR.	<ul style="list-style-type: none">• Recourir à des exemples de BDR touchant le vécu de l'apprenant.• Réviser et consolider les notions clés abordées les années précédentes via des activités/exercices issues de la vie courante de l'apprenant en relation avec :<ul style="list-style-type: none">- la définition et l'utilité d'une BD.- la définition et les fonctionnalités d'un Système de Gestion d'une Base de Données (SGBD).- la structure d'une BDR (table, relation, enregistrement et champ).- les contraintes d'intégrité.• Adopter, pour la représentation textuelle et graphique d'une BDR, la convention suivante :<ul style="list-style-type: none">- clé primaire soulignée,- clé étrangère suivie du symbole #.

		<ul style="list-style-type: none"> • Prévoir des activités construites autour de représentation (textuelle/graphique) erronée d'une BDR afin d'amener l'apprenant à : <ul style="list-style-type: none"> - détecter la(les) anomalie(s), - proposer une correction pour répondre aux contraintes d'intégrités définies.
	<ul style="list-style-type: none"> • Manipuler la structure d'une BDR en mode SQL. <ul style="list-style-type: none"> - Utiliser des requêtes pour créer une BDR et des tables. - Utiliser des requêtes pour modifier la structure d'une table (colonne et contrainte). - Utiliser des requêtes pour supprimer des tables et des BDR. • Manipuler des données d'une BDR en mode SQL. <ul style="list-style-type: none"> - Utiliser des requêtes pour insérer des données. - Utiliser des requêtes pour supprimer des lignes. - Utiliser des requêtes pour mettre à jour des lignes/colonnes. • Interroger une BDR en mode SQL. <ul style="list-style-type: none"> - Exploiter des requêtes mono-table. - Exploiter des requêtes avec jointures. - Exploiter des requêtes imbriquées. - Exploiter des requêtes en utilisant des fonctions agrégats et des groupements. 	<ul style="list-style-type: none"> • Adopter le standard SQL pour exploiter une BDR. • Prévoir des activités traitant l'importation et l'exportation au format SQL : <ul style="list-style-type: none"> - des données d'une table/BD. - de la structure d'une BD. • Se limiter aux sous-requêtes non corrélées dans la clause WHERE.

Systèmes, technologies et Internet	<p>➤ Exploiter des techniques de développement pour créer et publier un site web dynamique.</p> <ul style="list-style-type: none"> • Créer un site web interactif. <ul style="list-style-type: none"> - Créer des documents web en utilisant les éléments HTML5. <ul style="list-style-type: none"> ○ Intégrer l'élément <datalist> dans une page web. ○ Exploiter les événements HTML pour améliorer l'interactivité : onkeydown , onkeyup, onmouseover, onmouseout, onplay et onpause. 	<ul style="list-style-type: none"> • Exploiter un éditeur Web WYSIWYG qui intègre le HTML5. • Pour chaque élément html, traiter les attributs qui lui correspondent (voir annexe). • Réviser et consolider les notions clés abordées les années précédentes via des activités/exercices issues de la vie courante de l'apprenant en relation avec : <ul style="list-style-type: none"> - les éléments d'en-tête ; - les éléments de structuration sémantiques ; - les conteneurs ; - les éléments de texte ; - les éléments multimédias ; - les formulaires ; - les liens ; - le déclenchement d'évènements ; - la validation d'un document Web (HTML5).
	<ul style="list-style-type: none"> - Appliquer des styles CSS3 pour mettre en forme une page web. <ul style="list-style-type: none"> ○ Déclarer des sélecteurs CSS : sélecteurs de lien (:link, :visited), d'état (:hover, :focus) et d'attributs (élément[attribut], élément[attribut=valeur]). ○ Mettre en forme un document web en utilisant des sélecteurs CSS adaptés. ○ Ajouter un effet de transformation à un élément. 	<ul style="list-style-type: none"> • Réviser et consolider les notions clés abordées les années précédentes via des activités/exercices issues de la vie courante de l'apprenant en relation avec : <ul style="list-style-type: none"> - l'utilisation d'une feuille de style et d'une règle CSS ; - la création d'un document web en respectant la disposition HTML5 ;

		<ul style="list-style-type: none"> - la mise en forme d'un(e) texte, arrière-plan, image, liste, tableau, formulaire, conteneur et lien ; - l'application d'un effet de transition ; - l'utilisation des sélecteurs <u>d'élément</u>, <u>d'identifiant</u>, de <u>classe</u>, de <u>groupe</u> et <u>universel</u>. - la validation d'un document Web (CSS3).
	<ul style="list-style-type: none"> - Utiliser le langage JavaScript pour ajouter de l'interactivité dans un site web. <ul style="list-style-type: none"> o Exploiter les structures de contrôle switch et while. o Accéder à un élément du formulaire via son nom avec getElementByName. o Modifier le contenu, les attributs et les styles des éléments d'une page. 	<ul style="list-style-type: none"> • Le script JS doit être stocké dans un fichier externe. • Réviser et consolider les notions clés abordées les années précédentes via des activités/exercices issues de la vie courante de l'apprenant en relation avec : <ul style="list-style-type: none"> - l'utilisation des variables (déclaration et portée) ; - l'utilisation des objets String, Array, Math, Date et Number ; - l'utilisation des instructions d'affichage (innerHTML, write et alert) ; - l'utilisation des instructions d'entrée en utilisant l'invite (prompt) et les champs d'un formulaire ; - la récupération des données d'un formulaire en utilisant la méthode getElementById.

		<ul style="list-style-type: none"> - l'exploitation des structures de contrôle (if, for et do ... while); - l'utilisation des fonctions (déclaration et appel). • Inciter les apprenants à choisir les structures de données et de contrôle appropriés. • Le contrôle d'un champ d'un formulaire doit être effectué en JS sans utiliser l'attribut Pattern.
	<ul style="list-style-type: none"> • Créer un site Web dynamique <ul style="list-style-type: none"> - Reconnaître le principe de fonctionnement d'un site web dynamique. - Exploiter un environnement de travail. - Manipuler le langage de script PHP : <ul style="list-style-type: none"> ○ Identifier la structure de base d'un script PHP. ○ Manipuler des variables de différents types : entier, réel, booléen, chaîne de caractères, tableau indicé et tableau associatif. ○ Exploiter l'instruction d'affichage (echo). ○ Utiliser la structure de contrôle conditionnelle if. ○ Utiliser les structures de contrôle itératives : for, while et do...while. ○ Importer le contenu d'un fichier php dans un autre à l'aide de l'instruction require. 	<ul style="list-style-type: none"> • Prévoir des traitements sur les objets string et array. • Prévoir des activités pour la manipulation de la date et l'heure.

	<ul style="list-style-type: none">- Assurer la transmission des données entre des pages web.<ul style="list-style-type: none">○ Reconnaître le principe de transmission de données entre des pages web.○ Définir les variables superglobales (\$_GET et \$_POST).○ Utiliser les variables \$_GET et \$_POST.○ Transmettre des données via :<ul style="list-style-type: none">○ une URL en utilisant \$_GET.○ un formulaire en utilisant \$_POST.- Assurer l'interaction avec une BD :<ul style="list-style-type: none">○ Se connecter à un serveur de base de données.○ Sélectionner une base de données.○ Écrire et exécuter des requêtes SQL pour interroger et mettre à jour les données d'une base de données : SELECT, INSERT, DELETE et UPDATE.○ Exploiter les résultats d'une requête SQL pour les intégrer dans une page web.	<ul style="list-style-type: none">● Habituer les apprenants à exploiter, en PHP, des requêtes SQL générées en mode assisté.
--	---	---



Annexe HTML5

Eléments de 1^{er} niveau

<html>	Représente la racine d'un document HTML.
<head>	Conteneur de métadonnées placé entre la balise <html> et la balise <body>.
<body>	Définit le corps du document.

Eléments d'en-tête

<link>	Définit la relation entre le document actuel et une ressource externe. <i>Attributs : href, rel, type</i>
<meta>	Définit les métadonnées du document. <i>Attributs : charset, name, content</i>
<script>	Intègre un script côté client (JavaScript). <i>Attributs : src, type</i>
<style>	Définit les informations de style (CSS) pour un document. <i>Attributs : type, media</i>
<title>	Définit le titre du document.

Eléments de structuration d'une page web

<header>	Conteneur pour le contenu d'introduction ou un ensemble de liens de navigation.
<nav>	Définit un ensemble de liens de navigation.
<footer>	Définit un pied de page pour un document ou une section.
<section>	Définit une section dans un document.
<article>	Spécifie un contenu indépendant et autonome.
<aside>	Définit un contenu en dehors du contenu principal. Souvent placé sous forme de barre latérale.
<main>	Définit le contenu principal d'un document.

Eléments de structuration de texte et média

<cite>	Définit le titre d'une œuvre.
<p>	Définit un paragraphe.
<source>	Spécifie plusieurs ressources multimédias pour les éléments multimédias. <i>Attributs : src, type, media</i>
<h_n>	Définit un titre de niveau n (de 1 à 6).
<hr>	Définit une rupture thématique sous forme de ligne horizontale.
	Intègre une image. <i>Attributs : src, alt</i>
<figure>	Spécifie un conteneur autonome qui peut contenir divers éléments.
<figcaption>	Définit une légende pour un élément <figure>.
<audio>	Intégrer du contenu sonore dans un document. <i>Attributs : controls, src</i>
<video>	Intègre du contenu vidéo dans un document. <i>Attributs : controls, src</i>
<a>	Définit un lien hypertexte via l'attribut href. <i>Attributs : href, target</i>

	Définit un retour à la ligne.
<address>	Définit les coordonnées de l'auteur/propriétaire d'un document ou d'un article.
<output>	
<mark>	Définit le texte qui doit être marqué ou mis en surbrillance.
<details>	Spécifie des détails supplémentaires que l'utilisateur peut ouvrir et fermer à la demande. <i>Attribut : open</i>
<summary>	Définit un en-tête visible pour l'élément <details>.

Conteneurs

	Regroupe des éléments dans un bloc de contenu.
<div>	Grouper plusieurs éléments Html de type block.
<iframe>	Définit un cadre en ligne. <i>Attributs : src, name</i>

Listes

	Définit une liste à puces.
	Définit une liste ordonnée. <i>Attributs : reversed, type, start</i>
	Définit un élément dans une liste.

Tableau

<table>	Définit un tableau.
<thead>	Définit l'entête d'un tableau.
<tbody>	Définit le corps d'un tableau.
<tfoot>	Définit le pied d'un tableau.
<caption>	Définit une légende de tableau.
<tr>	Définit une ligne dans un tableau.
<th>	Définit une cellule d'en-tête dans un tableau.
<td>	Définit une cellule dans un tableau.

Evènements

onblur	Se déclenche au moment où l'élément perd le focus.
onclick	Se déclenche lors d'un clic sur l'élément.
onfocus	Se déclenche au moment où l'élément obtient le focus.
oninput	Se déclenche dès que la valeur d'un élément a changé.
onload	Se déclenche lorsque la page est complètement chargée.
onmouseover	Se déclenche lorsque le pointeur de la souris survole l'élément.
onmouseout	Se déclenche lorsque le pointeur de la souris sort de l'élément.
onplay	Se déclenche lorsque la lecture d'un fichier audio/vidéo commence.
onpause	Se déclenche lorsqu'un élément audio/vidéo est mis en pause.
onsubmit	Se déclenche lorsqu'un formulaire est soumis.
onchange	Se déclenche lorsque la valeur d'un élément est modifiée.
onkeydown	se déclenche lorsqu'une touche du clavier est pressée.
onkeyup	Se déclenche lorsqu'une touche est relâchée.

Attributs globaux

class	Spécifie un nom de classe pour un élément.
hidden	Renseigne la visibilité d'un élément.
id	Spécifie un identifiant unique pour un élément HTML.
lang	Spécifie la langue du contenu de l'élément.
style	Spécifie un style en ligne.
title	Spécifie des informations supplémentaires sur un élément.

Éléments de formulaire

		Attribut	Valeurs possibles	
<form>	Crée un formulaire HTML.	<i>action</i>		
		<i>method</i>	"get", "post"	
		<i>target</i>	"_self", "_blank", "_parent", "_top", "iframeName"	
<fieldset>	Regroupe des éléments liés dans un formulaire.			
<legend>	Définit une légende pour l'élément <fieldset>.			
<label>	Définit un libellé pour un élément graphique.			
<input>	Définit un champ de saisie.	Types d'élément	Attributs génériques	Attributs spécifiques
		button, reset, submit	<i>name, value, disabled</i>	
		text, email, tel	<i>name, value, placeholder, readonly, required, disabled, list</i>	
		password	<i>name, value, placeholder, readonly, required, disabled</i>	
		number, range	<i>name, value, readonly, required, disabled</i>	min, max
		date, time	<i>name, value, required, disabled</i>	min, max
		checkbox, radio	<i>name, value, required, disabled</i>	Checked
<textarea>	Définit une zone de saisie de texte multilignes.	Attributs : <i>name, cols, rows, readonly, required, disabled, placeholder, maxlength.</i>		
<select>	Définit une liste déroulante.			
<datalist>	Spécifie une liste d'options prédéfinies connectée à un élément <input>.			
<option>	Définit une option dans une liste de sélection.	Attribut : <i>selected</i>		



Annexe CSS3

Sélecteurs	
*	Sélectionne tous les éléments.
element	Sélectionne tous les éléments de type <i>element</i> .
#id	Cible un élément en fonction de la valeur de son attribut <i>id</i> .
.class	Cible les éléments en fonction de la valeur de leur attributs <i>class</i> .
element[attr]	Cible les éléments qui possèdent l'attribut <i>attr</i> .
element[attr=valeur]	Cible les éléments en fonction de la valeur de l'attribut <i>attr</i> .
element.class	Cible tous les éléments de type <i>element</i> en fonction de la valeur de leur attributs <i>class</i> .
a:link	Cible tous les liens non visités.
a:visited	Cible tous liens visités.
a:hover	Cible l'élément au moment où l'utilisateur le survole avec le pointeur de la souris.
a:active	Cible tous les liens activés.
Remarque: pour grouper plusieurs sélecteurs dans une règle, il suffit de les lister en les séparant par des virgules « , ».	

Propriétés de mise en forme du texte	
font-family	Spécifie les noms de polices possibles par ordre de préférence.
font-weight	Spécifie la manière dont les caractères d'un texte doivent être affichés (bold ; bolder ; lighter).
font-style	Spécifie le style d'un texte (italic).
font-size	Spécifie la taille d'une police.
font	Super-propriété de font. Combine font-family, font-weight, font-style, font-size.
text-align	Spécifie l'alignement horizontal d'un texte (left ; center ; right ; justify).
text-shadow	Définit une ombre au texte.
text-transform	Spécifie la façon de transformer les caractères d'un texte en majuscules ou en minuscules (uppercase ; lowercase ; capitalize).
color	Spécifie la couleur du texte.

Propriétés des listes	
list-style-type	Définit le type de marqueur d'éléments de liste (circle ; square ; upper-roman ; lower-alpha).
list-style-position	Spécifie la position des marqueurs d'éléments d'une liste à puces (outside ; inside).
list-style-image	Spécifie une image comme marqueur d'éléments de liste.
list-style	Super-propriété de liste. Combine list-style-type, list-style-position, list-style-image.

Propriétés d'arrière-plan	
background-color	Définit la couleur d'arrière-plan d'un élément.
background-image	Définit une image d'arrière-plan pour un élément.
background-repeat	Définit la façon dont une image d'arrière-plan est répétée (repeat ; repeat-x ; repeat-y).
background-size	Spécifie la taille d'une image d'arrière-plan.
background	Super-propriété d'arrière-plan. Combine background-color, background-image, background-repeat, background-size.

Transformation	
transform	Applique un effet de transformation 2D ou 3D à un élément (rotate() , skew() , scale() , translate()).

Propriétés des images	
filter	Définit des filtres sur un élément (blur() ; grayscale() ; invert()).

Propriétés des tableaux	
table-layout	Définit la façon de disposer les cellules, lignes et colonnes d'un tableau.
border-collapse	Définit si les bordures des cellules sont fusionnées ou séparées (separate ; collapse).

Propriétés des boîtes	
width	Définit la largeur d'un élément.
height	Définit la hauteur d'un élément.
position	Spécifie le type de la méthode de positionnement utilisée pour un élément (absolute ; fixed ; relative ; static ; sticky).
float	Place/Fait flotter un élément dans un conteneur (left ; right ; none).
padding	Propriété abrégée qui définit la marge intérieure d'un élément des 4 cotés.
margin	Définit les marges d'un élément.
box-shadow	Ajoute des ombres à la boîte d'un élément.
display	Spécifie le comportement d'affichage d'un élément (inline ; block ; inline-block).
top	Spécifie la position verticale, par rapport au haut, d'un élément positionné.
bottom	Spécifie la position verticale, par rapport au bas, d'un élément positionné.
left	Spécifie la position horizontale, par rapport à la gauche, d'un élément positionné.
right	Spécifie la position horizontale, par rapport à la droite, d'un élément positionné.
overflow	Spécifie le comportement du contenu lorsqu'il déborde de la boîte d'un élément (visible ; hidden ; clip ; scroll ; auto).
opacity	Définit le niveau de transparence.

Propriétés des bordures	
border-color	Définit la couleur des bordures d'un élément.
border-style	Définit le style des bordures d'un élément.
border-radius	Définit le rayon des coins arrondis d'un élément.
border-width	Définit la largeur des bordures d'un élément (medium ; thin ; thick ; valeur en px).
border	Super-propriété de bordure. Combine border-color, border-style, border-width, border-radius.

Transition	
transition-delay	Spécifie le délai nécessaire avant que la transition commence.
transition-duration	Spécifie la durée de la transition.
transition-property	Spécifie les propriétés CSS sur lesquelles l'effet de transition sera appliqué.
transition	Super-propriété de transition. Combine transition-property, transition-duration et transition-delay.

Animation	
@keyframes:	Spécifie les états clés de l'animation.
animation-name	Spécifie le nom de l'animation.
animation-delay	Spécifie le délai nécessaire avant que l'animation commence.
animation-duration	Spécifie la durée de l'animation.
animation-iteration-count	Spécifie le nombre de répétition de l'animation.
animation-direction	définit la direction de l'animation
animation	Super-propriété de l'animation. Combine animation-name, animation-duration, animation-delay, animation-iteration-count et animation-direction.

Important : pour exprimer les mesures des dimensions, se limiter aux unités (pixel et %).



Annexe JS

Opérateurs logiques

&&	ET
	OU
!	NON

Opérateurs de comparaison

=	Egal
!=	Différent
>	Supérieur à
>=	Supérieur ou égal à
<	Inférieur
<=	Inférieur ou égal à

Opérateurs arithmétiques

+	Addition
-	Soustraction
*	Multiplication
/	Division
%	Reste de la division euclidienne

Méthodes pour afficher une sortie*

alert()	Affiche un message dans une boîte de dialogue.
Document.write()	Affiche directement dans le document HTML.

*Pour afficher/modifier un contenu dynamique, on peut utiliser la propriété **innerHTML** d'un élément HTML.

Méthode pour saisir une entrée

prompt()	Affiche une boîte de dialogue avec une zone de saisie.
-----------------	--

Fonctions/Méthodes JS

isNaN(ch)	Retourne true si l'argument ch n'est pas convertible en un nombre, sinon retourne false.
Number(ch)	Convertit une chaîne ch en un nombre*.
parseFloat(ch)	Convertit une chaîne ch en un réel.
parseInt(ch [b])	Convertit une chaîne ch en un entier exprimé dans la base b.
String(a)	Convertit la valeur a en une chaîne.

*Si la conversion n'est pas possible, la fonction retourne NaN.

Méthodes pour sélectionner des éléments HTML

document.getElementById()	permet de sélectionner un élément unique en fonction de son identifiant.
document.getElementsByName()	permet de sélectionner tous les éléments ayant le même nom.

Méthodes de l'objet Math (Math)

abs(x)	Retourne la valeur absolue.
sqrt(x)	Retourne la racine carrée.
round(x)	Retourne l'entier le plus proche.
trunc(x)	Retourne la troncature entière d'un nombre en retirant sa partie décimale.
random()	Retourne un réel aléatoire dans [0, 1[.

L'objet Array

length	Propriété qui représente le nombre d'éléments dans un tableau.
---------------	--

Pour créer un tableau, utiliser les littéraux de tableau [].

Les chaînes de caractères

+	Opérateur de concaténation.
ch.length	Propriété qui retourne la longueur de ch.
ch.indexOf(ch [p])	Retourne la position de la 1 ^{ère} occurrence de ch1 dans ch, en effectuant la recherche à partir de la position p (sinon -1).
ch.lastIndexOf(ch1 [p])	Retourne la position de la dernière occurrence de ch1 dans ch à partir de la position p.
ch.substring(d, f)	Retourne une sous-chaîne de ch de la position d à la position f (non incluse).
ch.replace(ch1, ch2)	Retourne une chaîne dans laquelle la 1 ^{ère} occurrence de ch1 dans ch est remplacée par ch2.
ch.toLowerCase()	Convertit tous les caractères de ch en minuscules.
ch.toUpperCase()	Convertit tous les caractères de ch en majuscules.
ch.trim()	Supprime tous les espaces existants au début et à la fin de ch.
String.fromCharCode(num, ..., numN)	Retourne une chaîne formée par la concaténation des résultats de conversion des codes ASCII passés en paramètres.

ch est un objet String

L'objet Date

new Date()	Retourne la date courante.
new Date(ch)	Retourne un objet Date à partir d'une chaîne de caractères ch représentant une date.
d.getDate()	Retourne le jour du mois (entre 1 et 31).
d.getMonth()	Retourne le numéro du mois (entre 1 et 11).
d.getFullYear()	Retourne la valeur de l'année sur 4 chiffres.
d.toString()	Retourne une chaîne de caractères représentant la date d.

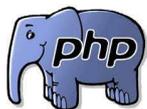
d est un objet Date.

Modification du contenu, des attributs ou des styles des éléments HTML

element.attribut = valeur	Change la valeur de l'attribut d'un élément HTML.
<i>Se limiter à la manipulation des attributs value, checked, disabled, readonly, src, muted.</i>	
element.style.propriété = valeur	Change le style d'un élément HTML.
<i>Se limiter à la manipulation des propriétés color, background, border, font, width, height.</i>	
element.méthode()	Effectue une action sur un élément HTML.

Se limiter à la manipulation des méthodes play(), pause().

Pour une liste déroulante (<select>), utiliser la propriété **selectedIndex** pour récupérer l'indice de l'option sélectionnée.



Annexe PHP

Type d'objets

int Entier

float Réel

string Chaîne de caractères

bool Booléen

array Tableau

Opérateurs de transtypage

(int) Convertit une variable en entier.

(float) Convertit une variable en réel.

(string) Convertit une variable en chaîne.

(bool) Convertit une variable en booléen.

(array) Convertit une variable en tableau.

Opérateurs logiques

&& Et

|| Ou

! Non

Opérateurs de comparaison

== égal à

<> différent de

< inférieur à

<= inférieur ou égal

> supérieur

>= supérieur ou égal

Opérateurs arithmétiques

+ addition

- soustraction

* multiplication

/ division

% modulo

Autres opérateurs

= affectation

· concaténation

Fonctions mathématiques

abs() Retourne la valeur absolue.

sqrt() Retourne la racine carrée.

round() Retourne la valeur arrondie.

rand() Retourne un nombre aléatoire.

Fonctions sur les chaînes de caractères

chr() Retourne le caractère dont le code ASCII est passé comme argument.

ord() Retourne le code ASCII du 1^{er} caractère de l'argument.

strlen() Retourne la longueur d'une chaîne.

substr() Retourne une sous-chaîne.

strpos() Retourne la position de la première occurrence du 2^{eme} argument dans le 1^{er}.

strcmp() Retourne le résultat de comparaison de deux chaînes (<0,0,>0).

str_replace() Remplace, dans une chaîne, toutes les occurrences d'un motif par un autre.

strtolower() Convertit tous les caractères d'une chaîne en minuscule.

strtoupper() Convertit tous les caractères d'une chaîne en majuscule.

trim() Supprime tous les espaces existants au début et à la fin de la chaîne.

Fonctions de Date/Heure

checkdate() Vérifie la validité d'une date.

date() Formate une date et la retourne sous forme de chaîne.

time() Retourne l'heure actuelle en nombre de seconde.

strtotime() Transforme une chaîne contenant une date en timestamp *

* timestamp: nombre de secondes écoulées entre une date donnée et le 1^{er} janvier 1970 à 00:00:00.

Fonctions sur les tableaux

array() Crée un tableau à partir d'une liste de valeurs.

count() Retourne le nombre d'éléments d'un tableau.

Structures de langage

echo Affiche une ou plusieurs expressions.

Fonctions PHP pour MySQL *

mysqli_connect() Établit une connexion à une base de données MySQL.

mysqli_query() Envoie une requête SQL à la base de données.

mysqli_close() Ferme la connexion à la base de données.

mysqli_fetch_array() Retourne une ligne de résultat sous forme de tableau associatif et numérique.

mysqli_fetch_row() Retourne une ligne de résultat sous forme de tableau indexé numériquement.

mysqli_num_rows() Retourne le nombre de lignes dans un résultat de requête.

mysqli_affected_rows() Retourne le nombre de lignes affectées par la dernière requête SQL exécutée.

mysqli_error() Retourne un message d'erreur généré par la dernière requête SQL exécutée.

* Utiliser les fonctions fournies par l'extension MySQLi dans son interface procédurale.

Variables superglobales

\$_GET Tableau associatif des valeurs passées au script courant via les paramètres d'URL.

\$_POST Tableau associatif des valeurs passées au script courant via le protocole HTTP et la méthode POST.

Fonctions diverses

die() Affiche un message et termine le script courant.

isset() Vérifie si une variable est définie.

require() inclut et exécute le fichier spécifié en argument.



Annexe SQL

Type de données

INT	Entier
DECIMAL	Réel
CHAR	Chaîne de caractères de longueur fixe.
VARCHAR	Chaîne de caractères de longueur variable dont la longueur maximale fixée.
TEXT	Chaîne de caractères de longueur variable.
DATE	Date.
TIME	Temps.
DATETIME	Date et Heure.

Opérateurs de comparaison

=	Égale à
<>	Différent de
>	Supérieur à
<	Inférieur à
>=	Supérieur ou égale à
<=	Inférieur ou égale à
IN	Dans
BETWEEN	Entre deux valeurs d'un intervalle.
LIKE	Recherche de motif/valeur.
IS	Filtre les résultats contenant la valeur NULL.

Opérateurs logiques

AND	ET
OR	OU
NOT	NON

Fonctions définies sur le type date

DATE()	Retourne une date extraite à partir d'une chaîne contenant une valeur au format DATE ou DATETIME.
DAY ()	Retourne le jour.
MONTH ()	Retourne le mois.
YEAR ()	Retourne l'année.
NOW ()	Retourne la date et l'heure courantes.
ADDDATE()	Ajoute un nombre de jours ou un intervalle de temps à une date.
DATEDIFF()	Retourne la différence entre deux dates en jours.

Fonctions d'agrégation

AVG ()	Retourne la moyenne.
COUNT ()	Retourne le nombre d'enregistrements.
MAX ()	Retourne le maximum.
MIN ()	Retourne le minimum.
SUM ()	Retourne la somme.

Manipulation des données

SELECT [DISTINCT] expression [, col, ... ou *] [[AS] alias] FROM table1 [[AS] alias] [, table2, ...] [WHERE condition] [GROUP BY critère] [HAVING condition] [ORDER BY expression [ASC/DESC]]	Spécification des colonnes du résultat. Spécification des tables Condition Organisation du résultat en groupes. Application d'un filtre. Tri du résultat.
INSERT INTO table [(liste colonnes)] VALUES (liste valeurs)	Insertion des données dans une table.
UPDATE table1 SET colonne = expression [WHERE condition]	Modification des valeurs d'une colonne.
DELETE FROM table1 [WHERE condition]	Suppression des enregistrements d'une table.

Contraintes d'intégrité

NOT NULL	Interdit une valeur nulle.
DEFAULT	Attribut une valeur par défaut.
CHECK	Spécifie que les valeurs de la colonne doivent respecter une condition.
PRIMARY KEY	Définit une clé primaire.
UNIQUE	Spécifie l'unicité des valeurs d'une colonne.
FOREIGN KEY	Définit une clé étrangère.
REFERENCES	Fait référence à une clé primaire d'une autre table.
ON UPDATE CASCADE	Met à jour la clé étrangère en cas de mise à jour de la clé primaire.
ON DELETE CASCADE	Supprime automatiquement tous les enregistrements de la table fille suite à la suppression de l'enregistrement correspondant de la table mère.

Définition des données

CREATE DATABASE nom	Crée une base de données.
DROP DATABASE nom	Supprime une base de données.
CREATE TABLE table1 (colonne1 type [contrainte], colonne2 type [contrainte], ... [[CONSTRAINT] [nom] [contrainte], ...)	Crée une table.
ALTER TABLE table1 ADD colonne type [contrainte]	Ajoute une colonne à une table.
ALTER TABLE table1 DROP colonne	Supprime une colonne.
ALTER TABLE table1 ALTER* colonne type	Modifie le type d'une colonne.
ALTER TABLE table1 RENAME** nom1 TO nom2	Modifie le nom d'une colonne.
ALTER TABLE table1 ADD CONSTRAINT [nom] contrainte	Ajoute une contrainte sur une colonne d'une table.
ALTER TABLE table1 DROP CONSTRAINT nom	Supprime une contrainte.
ALTER TABLE table1 ENABLE/ DISABLE CONSTRAINT nom	Active/Désactive une contrainte.
DROP TABLE table1	Supprime une table.

* on peut utiliser *MODIFY* au lieu de *ALTER*.

** on peut utiliser *CHANGE* au lieu de *RENAME*.