

## **LPW - Front-end Developer - Interrogation**

Nom : Adebayo  
Prénom : Samuel  
Durée : 1h00  
Date : 10/10/2024

### **Consigne :**

Le répertoire utilisé pour l'évaluation doit être compressé et envoyé au plus tard à 19h30 par e-mail à l'adresse **hamza.echamlali.4@formateur.ifapme.be**, via votre adresse e-mail étudiante.

### **Conditions :**

- L'utilisation de ChatGPT ou tout autre assistant IA est strictement interdite.
- Pas d'utilisation de frameworks JavaScript.
- Pas d'utilisation de frameworks CSS.

## **Exercice 1 : Création d'une page HTML5 complète (6 points)**

Créez une page HTML avec la structure suivante :

1. Le titre de la page (dans la balise <title>) doit être : "Banque : Compte d'Épargne".
2. Ajoutez une balise meta pour définir l'encodage de la page en UTF-8.
3. La page doit contenir une section principale (<section>) avec un en-tête (<header>) et le titre suivant (<h1>) : "Mon Compte d'Épargne".
4. Ajoutez un sous-titre (<h2>) sous l'en-tête avec le texte : "Gestion de Compte".
5. Sous le sous-titre, ajoutez un paragraphe expliquant : "Nous allons simuler des opérations bancaires simples en JavaScript."
6. Ajoutez une section nommée "Résultats". (Les informations concernant les opérations seront affichées uniquement dans la console, aucune sortie visible sur la page)

### **Évaluation :**

- Structure HTML correcte (2 points)
- Utilisation des balises de texte et d'en-tête (2 points)
- Ajout du paragraphe explicatif et de la section résultats (2 points)

## **Exercice 2 : Style de la page avec CSS3 (6 points)**

Ajoutez du style à la page avec un fichier CSS externe :

1. Appliquez la couleur de fond à la page de #f4f4f4.
2. L'en-tête (<header>) doit être centré avec une couleur de texte bleue foncée et utiliser la police Arial.
3. Ajoutez une bordure de 2px solide et gris clair (#dcdcdc) autour de la section principale.
4. Le paragraphe explicatif doit être en italique et centré horizontalement.
5. Modifiez la taille de police des titres (<h1> et <h2>) pour qu'ils soient respectivement de 28px et 20px.

### **Évaluation :**

- Style global de la page et centrage (2 points)
- Style des paragraphes et bordures (3 points)
- Taille correcte des titres (1 point)

## **Exercice 3 : Système de compte d'épargne en JavaScript (18 points)**

Dans un fichier JavaScript externe, réalisez les tâches suivantes pour simuler un compte d'épargne simple : *Les résultats doivent être affichés dans la console avec des console.log().*

1. Déclaration des variables et constantes :
  - Créez une constante appelée `tauxInteret` avec la valeur 0.03 (3% d'intérêt annuel).
  - Créez une variable appelée `solde` avec une valeur initiale de 1000 (solde initial en euros).
  - Créez une variable appelée `depot` avec la valeur 200 (montant à déposer sur le compte).
  - Créez une variable appelée `retrait` avec la valeur 150 (montant à retirer du compte).
2. Dépôt d'argent :
  - Ajoutez le montant de `depot` au `solde`.
  - Utilisez `console.log()` pour afficher le message : "Vous avez déposé 200 euros. Nouveau solde : X euros." où X est le nouveau solde après dépôt.
3. Retrait d'argent :
  - Vérifiez si le `solde` est suffisant pour permettre le retrait.
  - Si le `solde` est suffisant, soustrayez le montant de retrait du `solde`, puis affichez : "Vous avez retiré 150 euros. Nouveau solde : X euros."
  - Si le `solde` est insuffisant, affichez un message dans la console : "Solde insuffisant pour effectuer ce retrait."
4. Calcul des intérêts :
  - Multipliez le `solde` actuel par le `tauxInteret` et ajoutez le résultat au `solde`.
  - Utilisez `console.log()` pour afficher le message : "Intérêts annuels de 3% ajoutés. Nouveau solde : X euros."
5. Simulation de plusieurs opérations :
  - Créez une nouvelle variable appelée `nouveauDepot` avec une valeur de 500.
  - Ajoutez ce montant au `solde` et affichez un message similaire à celui utilisé pour le premier dépôt.
  - Créez une variable appelée `nouveauRetrait` avec une valeur de 800.
  - Effectuez un retrait en vérifiant que le `solde` est suffisant et affichez un message comme pour le premier retrait.
  - Enfin, recalculez les intérêts avec ce nouveau `solde` et affichez le résultat dans la console.

**Évaluation :**

- Déclaration correcte des variables et constantes (3 points)
- Ajout correct du dépôt et affichage (3 points)
- Retrait avec condition de solde suffisant (4 points)
- Calcul des intérêts annuels (3 points)
- Simulation de plusieurs opérations (5 points)

**Bonus (3 points) : Gestion de l'historique des opérations**

Ajoutez un système simple pour afficher l'historique des opérations dans la console. Créez un tableau vide appelé `historiqueOperations` dans lequel chaque opération sera enregistrée sous la forme de chaînes de caractères :

1. À chaque dépôt ou retrait, ajoutez une chaîne dans le tableau indiquant l'opération, par exemple : "Dépôt de 200 euros" ou "Retrait de 150 euros".
2. Après avoir effectué toutes les opérations, affichez le contenu du tableau `historiqueOperations` dans la console avec un `console.log()`.

**Évaluation Bonus :**

- Utilisation correcte du tableau pour l'historique (2 points)
- Affichage de l'historique dans la console (1 point)

**Note totale :** /30 + 3 points bonus