

TP PHP : Développement d'un Quiz interactif en mode Console

Niveau : BTS SIO - 1ère année

Durée estimée : 2 heures

Thème : Tableaux indexés, chaînes de caractères et interactions utilisateur.

1. Contexte du projet

Une entreprise de formation souhaite tester les connaissances générales de ses stagiaires en informatique via un petit outil léger et rapide. Votre mission est de développer un script PHP nommé quiz.php qui s'exécute dans la console (CLI).

Ce script posera une série de 20 questions à l'utilisateur, proposera 4 choix de réponses pour chacune, et calculera le score final.

2. Objectifs Pédagogiques

À la fin de ce TP, vous serez capable de :

- Parcourir un **tableau indexé**.
- Manipuler des chaînes de caractères (extraction de données via des séparateurs).
- Gérer les entrées utilisateur (User Input) en ligne de commande.
- Implémenter une logique de comptage de points.

3. Spécifications Techniques

A. Structure des données

Les données sont stockées directement dans le code via un tableau indexé.

Chaque élément du tableau est une chaîne de caractères (string) contenant toutes les informations d'une question, séparées par le caractère pipe |.

Format d'une ligne :

"Intitulé de la question|Réponse A|Réponse B|Réponse C|Réponse D|Numéro de la bonne réponse"

Exemple :

"Que signifie PHP |Personal Home Page|Private Home Page|Hypertext Preprocessor|Public Hyperloop Protocol|3"

(Ici, la bonne réponse est la 3ème proposition : Hypertext Preprocessor).

B. Fonctionnement du programme

1. **Message de bienvenue** : Le programme salue l'utilisateur.
2. **Boucle de jeu** : Le programme parcourt les 20 questions. Pour chaque question :
 - Il affiche l'intitulé.
 - Il affiche les 4 choix possibles (numérotés de 1 à 4).
 - Il demande à l'utilisateur de saisir son choix.
 - Il vérifie si la réponse est correcte.
3. **Score** :
 - Une bonne réponse rapporte **1 point**.
 - Une mauvaise réponse ne rapporte aucun point (et n'en retire pas).
4. **Fin de partie** : Le programme affiche le score final sur 20 avec un message personnalisé (ex: "Bravo !", "Peut mieux faire...").

C. Exemple d'affichage

=====


BIENVENUE DANS LE QUIZ BTS SIO

=====

Question 1 : Quel langage est principalement utilisé pour le style des pages web ?

- 1) HTML
- 2) CSS
- 3) Python
- 4) SQL


Votre réponse (1-4) : 1

 Mauvaise réponse... La bonne était la 2 (CSS)

Question 2 : Que signifie le sigle WWW ?

- 1) World Wide Web
- 2) Web World Wide
- 3) Wide Web World
- 4) World Web Wide

Votre réponse (1-4) : 1

 Bonne réponse !

... Suite du Quiz

=====

TERMINE ! Votre score final : 15 / 20

Validation acquise. Continuez vos efforts !

=====

4. Travail à réaliser

Étape 1 : Initialisation des données

Récupérer le script quiz.php avec le tableau déjà initialisé.

Étape 2 : Traitement et Affichage

Utilisez une boucle (type foreach ou for) pour parcourir le tableau \$questions.

À l'intérieur de la boucle, vous devrez utiliser la fonction native PHP explode() pour découper la chaîne de caractères et récupérer les différentes parties (Question, Choix, Réponse correcte).

Étape 3 : Interaction Utilisateur

Utilisez la fonction readline() pour récupérer la saisie de l'utilisateur.

Exemple : \$reponseUtilisateur = readline("Votre choix (1-4) : ");

Étape 4 : Vérification et Score

Comparez la réponse de l'utilisateur avec la bonne réponse extraite du tableau. Incrémentez une variable \$score si nécessaire.

Étape 5 : Affichage Final

Affichez le résultat final de manière claire.

Partie 2 : Évolution vers un "Mode Examen"

Objectif : Modifier la structure logique du script pour séparer la **saisie** des réponses de l'**analyse** des résultats.

1. Contexte de l'évolution

Dans la version précédente, l'utilisateur savait immédiatement s'il avait juste ou faux. Pour simuler un véritable examen, nous allons changer ce comportement. Désormais, l'utilisateur répondra aux 20 questions à la suite sans interruption. Ce n'est qu'à la fin que le score s'affichera, suivi d'un récapitulatif détaillé de ses erreurs.

2. Nouvelles Spécifications Techniques

A. Stockage des réponses

Vous ne pouvez plus traiter le résultat (vrai/faux) directement dans la boucle de question.

- Vous devez créer un nouveau tableau vide (par exemple `$reponsesUtilisateur`) au début du script.
- À chaque tour de boucle (lorsque l'utilisateur saisit un chiffre), vous devez **stocker** ce choix dans ce nouveau tableau.

B. Déroulement du script

Le programme doit désormais suivre **deux phases distinctes** :

Phase 1 : L'examen

- Parcourir les questions.
- Afficher l'intitulé et les choix.
- Enregistrer la réponse de l'utilisateur.
- *Ne rien afficher concernant la validité de la réponse (pas de "Bravo" ni de "Mauvaise réponse").*

Phase 2 : Le bilan (Après la fin des 20 questions)

- Calculer le score en comparant le tableau `$questions` et votre tableau `$reponsesUtilisateur`.
- Afficher la note finale.
- Afficher un **Récapitulatif Détaillé** ligne par ligne.

C. Format du récapitulatif attendu

Le récapitulatif doit permettre à l'utilisateur de comprendre ses erreurs. Pour chaque question, affichez une ligne formatée ainsi :

- Si la réponse était bonne : `[V] Question X : Bonne réponse !`
- Si la réponse était fausse : `[X] Question X : Vous avez répondu "A" mais la bonne réponse était "C" (Texte de la réponse).`

D. Exemple d'affichage

```
=====
      BILAN DES RÉSULTATS
=====
```

✓ Question 1 : Bonne réponse !

✓ Question 2 : Bonne réponse !

✓ Question 3 : Bonne réponse !

✓ Question 4 : Bonne réponse !

✗ Question 5 : Erreur.
-> Vous avez répondu : 443
-> La bonne réponse était : 80

... suite de l'affichage

✓ Question 18 : Bonne réponse !

✓ Question 19 : Bonne réponse !

✓ Question 20 : Bonne réponse !

```
=====
NOTE FINALE : 18 / 20
=====
```

Validation acquise. Continuez vos efforts.