

Explication

5)

foreach (\$t as &\$v) parcourt **par référence**, donc il double directement les valeurs : 10→20 et 20→40, le tableau devient [20, 40].

Ensuite \$t[] = 5 ajoute 5 en fin de tableau (équivalent à array_push(\$t, 5) en PHP et à t.push(5) en JavaScript).

implode("-", \$t) affiche alors 20-40-5

7)

unset[1] retire "IT" du tableau car index tableau commence par 0. Il a de base initialement 3 iteam au tableau et maintenant après le unset on en compte 2;

15)

parcours le fichier films ou on transforme chaque ligne en tableau (explode).

Ex ligne 1 : devient [101,Matrix,Action]. En suite , pour chaque ligne si le tableau position 2 = Action. On affiche le tableau position 1. Donc pour [101,Matrix,Action] on affiche Matrix.

On fait cela pour chaque ligne dans le fichier

17)

Piège : unset(\$f) supprime la variable temporaire, pas l'élément du tableau.

peut supprimer si reference exacte comme unset(\$film[i]);

ou i est valeur index tel que 0,1,2,3,.....n

18)

L'opérateur + garde les valeurs du **premier tableau** quand les clés sont identiques, tandis que **array_merge()** remplace ces valeurs par celles du **deuxième**.

Donc ici : r1["B"]=2 (pas écrasé) et r2["B"]=3 (écrasé)

19)

Sans saut de ligne avant le premier texte, tout reste sur la même ligne ; PHP_EOL n'en crée qu'un seul à la fin.

(Exemple contenu fichier : Bonjourle monde\n → une seule ligne)

20)

sort() réindexe toujours à partir de 0, donc les clés d'origine sont perdues.

21)

== convertit "0" en faux, mais === exige le même type (string ≠ bool).

22)

En PHP, la fonction header("Location: ...") envoie un en-tête HTTP demandant au navigateur de se rediriger vers une autre page. Cependant, **elle n'interrompt pas automatiquement le script** : le reste du code continue de s'exécuter.

Ainsi, dans ce code, la ligne echo "Film ajouté !"; peut encore s'afficher juste avant que la redirection s'effectue, provoquant parfois un message visible ou une erreur d'en-tête ("headers already sent").

La bonne pratique est donc de toujours ajouter exit; ou die; juste après la redirection pour **stopper le script immédiatement** :

23)

sans trim(), \$t[2] contient "Action\n", donc la comparaison échoue.

24)

array_search("A") renvoie 0 (premier index) qui est évalué **faux**

25)

Avec le **3^e paramètre true**, la comparaison est stricte ("1" ≠ 1).

26)

Le message est passé dans l'URL (?msg=ajoute), donc il est récupéré avec la **variable superglobale \$_GET**, pas \$_POST.