

PHP : Conditions - Exercices 🎧

Ces exercices te permettront de pratiquer les structures conditionnelles en PHP, avec et sans opérateurs (comparaison, logiques). Pour chaque exercice, écris ton script PHP, teste-le en CLI ou via ton serveur local, et vérifie que tous les cas sont bien pris en compte.

1. Vérification de la majorité

Nom du fichier: `conditions_01_majorite.php`

Objectif : Écrire un script qui demande à l'utilisateur son âge et affiche « Vous êtes majeur » si l'âge est ≥ 18 , sinon « Vous êtes mineur ».

- Sans opérateurs logiques (simple `if/else`). - Affichage dynamique du nombre saisi.

2. Pair ou impair

Nom du fichier: `conditions_02_pair.php`

Objectif : Créer un script qui génère un nombre aléatoire entre 1 et 100, puis affiche s'il est pair ou impair.

- Utilise l'opérateur modulo `%`. - Deux branches : `if (... % 2 == 0)` et `else`.

3. Note scolaire

Nom du fichier: `conditions_03_note.php`

Objectif : Demander une note entre 0 et 20, puis afficher :

- « Échec » si < 10 , - « Passable » si ≥ 10 et < 14 , - « Bien » si ≥ 14 et < 17 , - « Excellent » si ≥ 17 . - Utilise une chaîne de `if...elseif...else`.

4. Borne de température

Nom du fichier: `conditions_04_borne.php`

Objectif : Saisir une température en °C et afficher :

- « Trop froid » si < 0 , - « Frais » si ≥ 0 et ≤ 15 , - « Tempéré » si > 15 et ≤ 25 , - « Chaud » si > 25 . - Combine comparaisons simples, sans opérateurs logiques complexes.

5. Authentification basique

Nom du fichier: `conditions_05_auth.php`

Objectif : Créer un script qui compare deux variables `$user` et `$pass` à des valeurs prédéfinies.

- Si ``$user === "admin" ET ` $pass === "secret"`, afficher « Accès autorisé », sinon « Accès refusé ». - Utilise l'opérateur logique ``&&``.

6. Année bissextile

Nom du fichier: `conditions_06_annee.php`

Objectif : Vérifier si une année saisie est bissextile :

- Bissextile si divisible par 4 ET (pas divisible par 100 OU divisible par 400). - Exemples de tests : 2000, 1900, 2024, 2023. - Utilise ``&&``, ``||`` et les parenthèses pour bien ordonner les tests.

7. Tarif réduit

Nom du fichier: `conditions_07_tarif.php`

Objectif : Selon l'âge de l'utilisateur :

- Moins de 12 ans → tarif enfant,
 - 12 à 17 ans → tarif réduit,
 - 18 à 64 ans → tarif normal,
 - ≥ 65 ans → tarif senior.
 - Implémente avec `switch(true)` ou `if/elseif`.
-

8. Validation d'e-mail simple

Nom du fichier: `conditions_08_email.php`

Objectif : Demander une adresse e-mail et vérifier qu'elle contient un « @ » et un « . ».

- Sans utiliser de regex : ``if (strpos($email, "@") !== false && strpos($email, ".") !== false)``. - Affiche « Valide » ou « Invalide ».

9. Comparaison de chaînes

Nom du fichier: `conditions_09_chaines.php`

Objectif : Comparer deux chaînes saisies par l'utilisateur :

- Si elles sont identiques (opérateur `===`), afficher « Identiques », sinon « Différentes ». - Teste la sensibilité à la casse.

10. Calculatrice basique

Nom du fichier: `conditions_10_calc.php`

Objectif : Créer un script où l'utilisateur saisit deux nombres et un opérateur (+, -, *, /).

- Utilise `switch` sur l'opérateur pour effectuer le calcul. - Gère le cas division par zéro avec un `if` avant d'exécuter `.`.

Ce qu'il faut retenir

- Les conditions `if`, `elseif`, `else` te permettent de diriger le flux de ton programme selon des tests simples ou multiples.
- Les opérateurs de comparaison (`==`, `===`, `<`, `>`, `<=`, `>=`) et logiques (`&&`, `||`, `!`) offrent une grande flexibilité pour exprimer des conditions complexes.
- Le `switch` peut être une alternative élégante à une chaîne de `if` pour des tests sur une même variable.
- Toujours tester tous les cas possibles (valeurs limites, erreurs d'utilisateur) pour éviter les comportements inattendus.