PHP : switch

Le `switch` te permet de tester plusieurs valeurs possibles d'une même variable de façon claire et organisée, sans enchaîner de nombreux `if…elseif`.

Objectifs de la leçon

- Comprendre la syntaxe du `switch`.
- Savoir l'utiliser pour remplacer des chaînes de `if…elseif`.
- Voir un exemple concret d'utilisation.

Syntaxe de base

```php
switch ($variable) {
    case valeur1:
        // instructions si $variable == valeur1
        break;
    case valeur2:
        // instructions si $variable == valeur2
        break;
    // …
    default:
        // instructions si aucune valeur ne correspond
}
```

- **`$variable`** : l'expression à tester.
- **`case valeurX:`** : chaque branche testée avec l'opérateur `==`.
- **`break;`** : termine la branche pour éviter d'exécuter les suivantes.
- **`default:`** : optionnel, s'exécute si aucun `case` n'est vrai.

Exemple concret : Menu Utilisateur

Imaginons un script qui affiche une page selon le rôle d'un utilisateur.

```php
<?php
$role = "éditeur"; // valeur récupérée dynamiquement

switch ($role) {
    case "administrateur":
        echo "Bienvenue, administrateur !";
        break;
    case "éditeur":
        echo "Bienvenue, éditeur : vous pouvez modifier le contenu.";
        break;
    case "abonné":
```

```php
        echo "Bienvenue, abonné : accès en lecture seule.";
        break;
    default:
        echo "Rôle inconnu : accès limité.";
}
?>
```

Explications :

1. Le rôle est comparé à chaque `case`.
2. À la correspondance, les instructions du `case` s'exécutent.
3. `break` empêche la « chute » vers les cases suivantes.
4. `default` gère les rôles non prévus.

Bonnes pratiques

1. **Toujours inclure `break`** dans chaque `case` sauf si tu veux volontairement enchaîner plusieurs cas.
2. **Penser au `default`** pour traiter les valeurs inattendues.
3. **Choisir `switch`** plutôt que plusieurs `if…elseif` quand tu testes une seule variable sur plusieurs valeurs.
4. **Garder chaque `case` simple** : évite d'y mettre trop de logique ; préfère appeler une fonction si nécessaire.
5. **Commentaires** : nomme clairement chaque `case` pour faciliter la relecture.

Ce qu'il faut retenir

- Le `switch` teste une variable contre plusieurs valeurs.
- Chaque `case` se termine par `break`.
- Le `default` gère les cas non prévus.
- Idéal pour remplacer des chaînes complexes de conditions.
- Favorable à la lisibilité et à la maintenance du code.