

La dynamisation

Ou comment faire cohabiter les données et la mise en forme

Définition :

La dynamisation consiste à rendre **variable** les données affichées sur une page, dont la provenance peut-être : une base de données, un fichier donné, un autre site... On dit alors que les données sont **dynamiques** lorsqu'elles changent en fonction du contenu de la source. L'inverse de données dites **dynamiques** sont les données dites **statiques**.

L'exemple le plus courant de **dynamisation** consiste à coupler du code HTML avec du code PHP. On demande alors au serveur de nous renvoyer le code HTML tel qu'écrit dans le fichier, mais EN PLUS d'exécuter le code PHP pour afficher les données ou pour traiter les boucles etc.

Exemple : Deux contenus identiques, l'un statique, l'autre dynamique

```
11 <body>
12   <div>
13     Nicolas
14   </div>
15   <div>
16     SAVOCA
17   </div>
18   <div>
19     TP DEV WEB
20   </div>
21   <div>
22     Moulins
23   </div>
24 </body>
25
```

```
<body>
<?php
$resultatBDD = array(
    "prenom" => "Nicolas",
    "nom" => "SAVOCA",
    "formation" => "TP DEV WEB",
    "ville" => "MOULINS"
);
foreach($resultatBDD as $champ => $valeur) {
    ?>
    <div>
        <?= $valeur ?>
    </div>
    <?php
}
?>
</body>
```

Le résultat de ces deux contenus va être identique, seulement l'un est statique et l'autre est dynamique. On demande au serveur d'exécuter du code PHP pour **générer** du code HTML.

Lorsque l'on intègre du code PHP dans du HTML, on a des fois besoin de gérer des algorithmes plus complexes comme des **foreach** ou des successions de **if**. Mais des fois on veut simplement afficher le contenu d'une variable.

Pour faire cela plus simplement, on peut utiliser la syntaxe

```
<?= $valeur ?>
```

C'est quand même plus court que :

```
<?php
echo $champ;
?>
```

Affichage conditionnel

En utilisant la synergie entre PHP et HTML, on peut donc gérer plusieurs affichages différents dans le même fichier HTML.

Par exemple, dans le cas d'une page d'accueil, on peut gérer un menu différent selon si l'utilisateur est connecté ou non. Pour cela, rien de tel que la syntaxe **if**.

Exemple :

Dans ce menu on a deux affichages différents délimités par une syntaxe **if / else**

```
<div class="container">
  <a href="/" class="logo">
    
  </a>
  <?php
  if(array_key_exists('id_utilisateur', $_SESSION)) {
    ?>
    <div class="dropdown">
      <button class="btn btn-primary dropdown-toggle" type="button"
data-bs-toggle="dropdown" aria-expanded="false">
        <i class="fa-solid fa-user"></i> Mon compte
      </button>
      <ul class="dropdown-menu">
        <li><a class="dropdown-item" href="/profil.php">Mon
profil</a></li>
        <li><a class="dropdown-item"
href="/?logout=1">Déconnexion</a></li>
      </ul>
    </div>
    <?php
  } else {
    ?>
    <!-- Ce dropdown s'affiche par défaut lorsque personne n'est connecté
-->
    <div class="dropdown">
      <button class="btn btn-primary dropdown-toggle" type="button"
data-bs-toggle="dropdown" aria-expanded="false">
        <i class="fa-solid fa-user"></i> Connexion
      </button>
      <ul class="dropdown-menu">
        <li><a class="dropdown-item"
href="/inscription.php">S'inscrire</a></li>
        <li><a class="dropdown-item" href="/connexion.php">Se
connecter</a></li>
      </ul>
    </div>
    <?php
  }
  ?>
```

```
</div>
```