

Développement d'un site web dynamique (UAA12)

Exercices : bases de PHP

Objets d'apprentissage :

- ✓ Construire une page WEB dynamique à l'aide du langage PHP.

Exercice 1 : types de valeurs et affichage

Etape 1.

Utilise l'outil en ligne <https://onlinephp.io/> pour réaliser cet exercice.

Exécute le script suivant et observe l'affichage :

```
1 <?php
2 $x = 0;
3 echo "\$x vaut $x et est de type " . gettype($x) . "\n";
4 echo "Var_export indique : ";
5 var_export($x);
6 echo "\nVar_dump indique : ";
7 var_dump($x);
8 echo "\n";
9 ?>
```

Comprends-tu toutes ces lignes de code ? Voici quelques rappels pour t'aider :

- [ligne 3] le symbole `.` permet de concaténer plusieurs chaînes de caractères
- [ligne 3] la fonction `gettype` permet d'obtenir le type d'une variable
- [ligne 5] la fonction `var_export` donne la valeur d'une variable
- [ligne 7] la fonction `var_dump` donne la valeur et le type d'une variable

Etape 2.

Dans les instructions `echo`, remplace les guillemets par des apostrophes et observe le résultat. Quelle est la différence entre les guillemets et les apostrophes en PHP ?

Reviens aux guillemets avant de poursuivre à l'étape suivante.

Etape 3.

Modifie ensuite la ligne 2 :

```
$x = 135.2;
```

Exécute à nouveau le script. Observe les résultats de l'affichage via `echo`, `var_export` et `var_dump`.

Essaie de nouveau avec les initialisations suivantes :

```
$x = true;
$x = false;
$x = "hello";
$x = null;
```

Observe la manière dont PHP affiche les valeurs booléennes !

Exercice 2 : des variables dans des chaînes de caractères

Crée une page PHP présentant les informations suivantes :

Je m'appelle Picsou.

J'adore l'argent.

J'en récolte environs \$3000 par mois c'est à dire \$36000 par an.

Dans ce texte, les parties soulignées sont variables. Leur contenu change selon quatre variables :

- **\$nom** : le nom de la personne (ici « Picsou »)
- **\$adore** : un booléen qui indique si la personne « adore » (`true`) ou « déteste » (`false`)
- **\$chose** : la chose que la personne adore ou déteste (ici « l'argent »)
- **\$salaire** : le salaire désiré en dollars (ici 3000)

Ces quatre variables doivent être initialisées au début du script PHP.

On te demande de créer quatre versions différentes de ce script :

1. Initialise les variables à l'aide des guillemets. Utilise l'HTML pour les parties fixes de chaîne de caractères (« Je m'appelle » par exemple) et le PHP pour les parties variables.
2. Idem mais remplace les apostrophes à la place des guillemets.
3. Fais en sorte que le contenu de la balise `<body>` soit produit par du code PHP en utilisant uniquement les apostrophes pour les chaînes de caractères.
4. Remplace les apostrophes par des guillemets et incorpore les variables dans les chaînes de caractères.

Exercice 3 : des étoiles...

On te demande d'écrire un script PHP qui, à partir d'une chaîne de caractères et d'un entier (stockés dans des variables), affiche une ligne répétant la chaîne de caractères autant de fois que demandé. Par exemple, si l'entier vaut 5 et que la chaîne de caractères vaut « * », la page affiche « ***** »

On te demande de réaliser cet exercice en deux versions différentes :

1. Avec une boucle **for** (comme en C et JavaScript) : l'affichage se fait au fur et à mesure (à chaque itération de la boucle)
2. Egalement avec une boucle **for** mais en construisant la chaîne de caractère progressivement et en l'affichant une seule fois (après la boucle)

Exercice 4 : ...et encore des étoiles

Fait une copie du script de l'exercice précédent et modifie-le pour obtenir le résultat suivant.

Le script doit afficher une ou plusieurs lignes d'étoiles selon un entier : une ligne pour chacun des chiffres de l'entier.

Par exemple, pour l'entier `$nb = 6934` le script doit afficher :

```
*****  
*****  
***  
****
```

Pour t'aider, voici deux fonctions pratiques :

- `strlen` : donne la taille d'une chaîne de caractères
(https://www.w3schools.com/php/func_string_strlen.asp)
- `substr` : retourne une partie d'une chaîne de caractères
(https://www.w3schools.com/php/func_string_substr.asp)

Exercice 5 : isset et empty

La fonction **isset** permet de vérifier si une variable existe. La fonction **empty**, elle, vérifie si une variable est vide. Mais la signification de « existe » et de « vide » n'est pas forcément très facile à définir.

Dans cet exercice, on te demande de compléter le tableau suivant en indiquant, pour chaque cas, la valeur booléenne renvoyée par chacun des tests.

Utilise le script suivant en remplaçant simplement la première ligne par chaque déclaration du tableau :

```
<?php
// Ligne à remplacer par une déclaration du tableau
echo isset($x);
echo empty($x);
?>
```

| | isset(\$x) | empty(\$x) |
|--------------------|------------|------------|
| Aucune déclaration | | |
| \$x = 0; | | |
| \$x = 0.0; | | |
| \$x = 7; | | |
| \$x = 27.4; | | |
| \$x = true; | | |
| \$x = false; | | |
| \$x = ""; | | |
| \$x = "\n"; | | |
| \$x = "hello"; | | |
| \$x = null; | | |
| \$x = array(); | | |

Références

Les présents exercices ont été élaborés à l'aide des ressources suivantes :

- Forbes, A. (2013). The Joy of PHP: A Beginner's Guide to Programming Interactive Web Sites.
- Vaswani, V. (2021). PHP A Beginner's Guide.
- Cours de « Développement d'applications WEB » (2018), Hénallux