Sujet 1 : Premiers pas en PHP
Buts : Types, Fonctions, Tableaux

Avant de commencer

Créez un répertoire `POO`, dans lequel vous créerez un sous-répertoire `TP1`. Vous y stockerez l'ensemble des développements produits tout au long de ce TP. Votre enseignant devra y retrouver la trace de tout le travail effectué, ne remplacez donc pas vos solutions par d'autres, privilégiez, en cas de nécessité, une mise en commentaire. Pour une question de lisibilité, veillez à reporter les numéros des questions dans votre code. De la même façon les noms des auteurs devront apparaître au début de chaque script.

Création du dépôt Git

Votre travail sera impérativement déposé sur un dépôt Git.

Consignes

Vous veillerez à respecter scrupuleusement les consignes suivantes :

- Vous effectuerez un commit après chaque exercice.
- A la fin de chaque séance, vous effectuerez un commit. Si ce commit contient du code incomplet ou ne fonctionnant pas, mentionnez-le dans le message de commit. Vous pousserez ensuite votre travail vers le dépôt distant.
- Ce dépôt sera utilisé par votre enseignant(e) de TP pour évaluer votre travail. Assurez-vous donc régulièrement que tous les fichiers que vous souhaitez lui rendre sont bien présents dans le dépôt.
- Le dépôt lui-même sera évalué : soignez l'écriture de vos messages.

Initialisation du dépôt

- Ouvrez un terminal et utilisez la commande *cd* pour vous placer dans le répertoire $TP1$
- Initialisez le dépôt *Git* du projet.
- Si nécessaire, éditez le fichier *.gitignore*
- Effectuez un premier commit
- Connectez-vous à l'application *Gitlab* et créez un dépôt nommé $POO - TP1$
- Assignez le rôle *Reporter* à votre enseignant(e) de TP.
- Poussez le dépôt local vers le dépôt distant.

Liens utiles

- Aide-mémoire Git : *https://iut-info.univ-reims.fr/users/nourrit/git_aide_memoire.pdf*
- Gitlab : *https://iut-info.univ-reims.fr/gitlab*

Les bons usages

D'après les recommandations PSR-2, un script PHP débute par la balise <?*php* mais ne se termine par aucune balise. Vous éviterez de fermer vos scripts avec la balise ? >.

Afin que le typage de vos fonctions et de vos méthodes soit réellement pris en compte nous procéderons dorénavant à un typage strict. Vous rajouterez pour ceci la directive suivante au début de vos scripts :

```
declare(strict_types = 1);
```

Premiers pas en PHP

Exercice 1 : les types de données

Créez un script PHP nommé *Exercice*1*.php* . Les réponses aux questions de l'exercice 1 viendront s'intégrer à ce script. Vous pourrez répondre (sous forme de commentaires) aux différentes questions de cours directement dans ce fichier.

Question 1.
Affichez la chaîne de caractères contenant vos nom et prénom suivis d'un passage à la ligne. Il ne vous est pas demandé d'utiliser des variables dans cette question.

Question 2.
En PHP quels sont les types de données correspondant à un entier, un réel et une chaîne de caractères. Créez une variable pour chacun de ces types puis affichez leurs valeurs respectives.

Question 3.
Créez une variable contenant la somme des variables précédemment créées. Affichez le résultat. Que constatez-vous ?

Question 4.
Créez deux variables de type `string` contenant la même chaîne mais l'une délimitée par des ′, l'autre par des ". Affichez-les. Que constatez-vous ?

Question 5.
Créez deux variables de type `string`. Créez une variable qui contient la somme des deux chaînes. Affichez la variable ainsi obtenue. Que constatez-vous ? *Remplacez* maintenant l'utilisation de l'opérateur + par celle de l'opérateur . (*point*). Affichez la variable ainsi obtenue. Que constatez-vous ?

Question 6.
Créez une variable $*entier*. Créez deux variables de type `string`, l'une délimitée par des ′, l'autre par des ". Ces deux chaînes devront contenir la variable $*entier* (l'appel à l'opérateur . n'est pas autorisée dans cette question). Affichez les deux chaînes de caractères. Que constatez-vous ?

Complétez vos deux chaînes caractères avec le caractère $\backslash n$. Affichez les deux chaînes de caractères. Que constatez-vous ?

Exercice 2 : les fonctions

Créez un script PHP nommé *Exercice*2*.php* . Les réponses aux questions de l'exercice 2 viendront s'intégrer à ce script. Vous pourrez répondre (sous forme de commentaires) aux différentes questions directement dans le fichier. Vous veillerez à tester toutes vos fonctions. Les fonctions que vous proposerez devront être typées : type des paramètres et type de retour. Un seul *return* sera toléré par fonction.

Question 7.
Définissez une fonction `afficheNomPrenom` qui prend en paramètre le nom et le prénom d'une personne et qui ne retourne rien. Cette fonction affichera les nom et prénom de la personne séparés d'un espace puis passera à la ligne.

Question 8.
Définissez une fonction `calculSomme` qui prend en paramètre trois entiers et retourne leur somme.

Question 9.
Définissez une fonction `getAge` qui ne prend rien en paramètre. Cette fonction permet de saisir au clavier l'age d'une personne et le retourne sous forme d'un entier. Un message explicite devra être affiché à l'utilisateur lors de la saisie. Vous pourrez pour ceci utiliser la fonction suivante :

```
readline ([ string $prompt ] ) : string
```

qui permet de saisir une ligne. Le message $*prompt* est ici optionnel. On supposera ici que l'utilisateur saisira des données cohérentes.

Question 10.
Définissez une fonction `plusGrand` qui prend en paramètre deux entier $*min* et $*max* et qui retourne un booléen. Cette fonction effectuera un tirage aléatoire de deux nombres entiers compris entre $*min* et $*max* inclus. Elle affichera les deux nombres puis retournera *true* si le premier entier est strictement plus grand que le second, $false$ sinon. On veillera à se limiter à une seule utilisation du *return*. Vous pourrez pour ceci utiliser la fonction suivante :

```
rand ( int $min , int $max ) : int
```

qui retourne un nombre aléatoire compris entre $*min* et $*max* inclus.

Vous veillerez à ne pas afficher une valeur booléenne directement. On pourra pour ceci définir une fonction `decodeBooleen` qui prend en paramètre un booléen et retourne la chaîne de caractères '$Vrai$' si le booléen a pour valeur *true*, '$Faux$' sinon. Cette fonction pourra être réutilisée ultérieurement.

Exercice 3 : les tests et boucles

Créez un script PHP nommé $Exercice3.php$. Les réponses aux questions de l'exercice 3 viendront s'ajouter à ce script. Vous pourrez répondre (sous forme de commentaires) aux différentes questions directement dans le fichier. Vous veillerez à tester toutes vos fonctions. On se limitra ici à l'utilisation des boucles *while* et for, le $foreach$ sera vu ultérieurement.

Question 11.
Listez les 8 opérateurs de comparaison qui existent en PHP et caractérisez-les.

Question 12.
Listez les 6 opérateurs logiques qui existent en PHP et caractérisez-les.

Question 13.
Le marchand de légumes en gros propose une réduction : 5% si le montant de la commande atteint ou dépasse 50 euros, 15% s'il atteint ou dépasse 100 euros, 25% s'il atteint ou dépasse 200 euros. Définissez une fonction `calculerRemise` prenant en paramètre le prix au kilo et le poids en grammes d'un article et retournant le prix tenant compte de la réduction.

Question 14.
Définissez une fonction `estBissextile` qui prend en paramètre une année et qui retourne booléen. Elle retourne un vrai si elle est bissextile, faux sinon. Pour rappel une année est bissextile si elle est divisible par 4 et non divisible par 100, ou divisible par 400. Vous veillerez à ne pas afficher une valeur booléenne retournée directement.

Question 15.
Définissez une fonction `auMoinsDeuxVrais` qui prend en paramètre trois booléens et retourne *true* si au moins deux de ces booléens sont vrais, et $false$ dans le cas contraire. On veillera à proposer une solution qui effectuera un minimum de comparaisons. Vous veillerez à ne pas afficher une valeur booléenne retourée directement.

Question 16.
Définissez une fonction `exactementDeuxVrais` qui prend en paramètre trois booléens et retourne *true* si exactement deux de ces booléens sont vrais, et $false$ dans le cas contraire. On veillera à proposer une solution qui effectuera un minimum de comparaisons. Vous veillerez à ne pas afficher une valeur booléenne retournée directement.

Question 17.
Il arrive souvent qu'une saisie doive respecter une consigne, par exemple répondre par oui ou par non. Tant que l'utilisateur ne respecte pas cette consigne, la question est reposée.
Définissez la fonction `saisieReponse` qui ne prend rien en paramètre, et qui permet de saisir la réponse de l'utilisateur. La saisie s'arrête lorsque l'utilisateur répond 'Oui' ou 'Non'. La réponse saisie par l'utilisateur sera retournée par la fonction.
Pour les plus avancés l'utilisation d'un *switch* permettrait de tolérer les réponses *oui*, *Oui*, OUI, o, O, Yes, yes, YES, y et Y pour le *Oui*, et des réponses aussi variées pour le Non

Question 18.
Définissez la fonction `dessinerRectanglePlein` qui prend en paramètre un motif (caractère), le nombre de lignes et le nombre de colonnes du rectangle. Cette fonction affichera un rectangle composé de $lignes \times colonnes$ motifs tous séparés par des espaces.

Exercice 4 : les tableaux indexés

Créez un script PHP nommé $Exercice4.php$. Les réponses aux questions de l'exercice 4 viendront s'ajouter à ce script. Vous pourrez répondre (sous forme de commentaires) aux différentes questions directement dans le fichier. Vous veillerez à tester toutes vos fonctions.

IMPORTANT : vous utiliserez la boucle for pour parcourir vos tableaux dans l'exercice 4. La boucle $foreach$ sera abordée dans l'exercice 5.

Question 19.
Définissez la fonction `creeTableau` qui prend en paramètre 3 entiers qui sont la $\$taille$ et les valeurs $\$min$ et $\$max$ contenues dans le tableau. Cette fonction permet de créer un tableau indexé d'entiers de $\$taille$ éléments tirés aléatoirement. Ces éléments auront des valeurs comprises entre $\$min$ et $\$max$ inclus. Cette fonction retournera le tableau créé. Afin de tester votre fonction pensez à utiiser `var_dump`.

Question 20.
Définissez la fonction `afficheTableau` qui prend en paramètre un tableau et permet d'afficher son contenu. L'affichage du tableau sera délimité par des [ ] et les éléments séparés par des espaces. On supposera ici que le tableau passé en paramètre est un tableau indexé.

Question 21.
Définissez la fonction `calculerMoyenne` qui prend en paramètre un tableau et retourne la moyenne des valeurs qu'il contient. On supposera ici que le tableau passé en paramètre est un tableau indexé d'entiers.

Question 22.
Définissez la fonction `estPresent` qui prend en paramètre un tableau et une valeur entière. Cette fonction retourne *true* si la valeur est présente dans le tableau, $false$ sinon. On supposera ici que le tableau passé en paramètre est un tableau indexé d'entiers.

Question 23.
Définissez la fonction `indexOf` qui prend en paramètre un tableau ainsi qu'une valeur entière. Cette fonction retourne l'indice de la première occurrence de la valeur dans le tableau, -1 sinon. On supposera ici que le tableau passé en paramètre est un tableau indexé d'entiers.

Question 24.
Définissez la fonction `getMaximum` qui prend en paramètre un tableau. Cette fonction retourne la valeur maximale contenue dans le tableau. On supposera ici que le tableau passé en paramètre est un tableau indexé d'entiers.

Question 25.
Définissez la fonction `getIndiceMinimum` qui prend en paramètre un tableau. Cette fonction retourne l'indice de la valeur minimale contenue dans le tableau. On veillera ici à ne pas stocker dans une variable le minimum du tableau. On supposera ici que le tableau passé en paramètre est un tableau indexé d'entiers.

Question 26.
En PHP, lorsqu'une fonction prend en paramètre un paramètre de type scalaire $float$, *int*, *bool* ou *string*, la modification de ce paramètre est-elle visible en dehors de cette fonction ? Justifiez votre réponse. Proposez un jeu de tests pertinent pour conforter votre argumentaire.

Question 27.
Définissez la fonction `echange` qui prend en paramètre un tableau ainsi que deux indices. Cette fonction inversera le contenu des cases correspondant aux deux indices, si ceux-là existent. On supposera ici que le tableau passé en paramètre est un tableau indexé d'entiers.

Afficher le tableau avant et après l'appel de la fonction échange, ainsi qu'à l'intérieur de la fonction `echange` avant et après échange. Le tableau a-t-il été réellement modifié ? Expliquez pourquoi. Vous pourrez vous inspirer de la réponse à la question précédente.

Exercice 5 : les tableaux associatifs (exercices complémentaires)

Créez un script PHP nommé *Exercice5.php* . Les réponses aux questions de l'exercice viendront s'ajouter à ce script. Vous pourrez répondre (sous forme de commentaires) aux différentes questions directement dans le fichier. Vous veillerez à tester toutes vos fonctions.

Question 28.
Définissez un tableau $*tableau* de la manière suivante :

```
$tableau = array ("un", "deux", "trois", "quatre");
```

Affichez le contenu de ce tableau en utilisant la boucle *for*. Chaque ligne contiendra l'indice suivi du caractère − et de la valeur correspondant à cet indice dans le tableau suivie d'un passage à la ligne.

Effectuez le même affichage en utilisant la boucle *foreach* dont la syntaxe générique est :

```
foreach ($tableau as $element) { ... }
```

A-t-on alors accès à l'indice dans le tableau ?

Effectuez le même affichage en utilisant la boucle *foreach* dont la syntaxe générique est :

```
foreach($tableau as $cle => $element) { ... }
```

A quoi correspondent ici les clés dans le tableau ?

Question 29.
Définissez un tableau $*tableau*2 de la manière suivante :

```
$tableau2 = array (1 => "un", "deux", "trois", "quatre");
```

Affichez le contenu de ce tableau en utilisant la boucle *for*. Chaque ligne contiendra l'*indice* suivi du caractère − et de la valeur correspondant à cet indice dans le tableau suivie d'un passage à la ligne.

Effectuez le même affichage en utilisant les *deux* boucles *foreach*.

A quoi correspondent ici les *indices* dans le tableau ?

Question 30.
Créez un tableau jour dans lequel la clé correspondra au numéro du jour de la semaine et la valeur au nom du jour. Le premier jour (jour 1) de la semaine sera "*Lundi*", le septième "*Dimanche*". Affichez le contenu de ce tableau en utilisant la boucle *foreach* la plus adaptée.

Définissez la fonction `determineJour` qui prend en paramètre le numéro du jour de la semaine ainsi que le tableau des jours de la semaine. Cette fonction retournera le nom du jour de la

semaine lorsque le numéro est compris entre 1 et 7, la chaine de caractères "*Inconnu*" sinon. Testez de manière exhaustive la fonction `determineJour` dans une boucle *for*.

Question 31.

Créez un tableau jour dans lequel la clé correspondra au nom du jour de la semaine en français et la valeur au nom du jour de la semaine en anglais. On y associera par exemple à "*Lundi*" la valeur "*Monday*". Affichez le contenu de ce tableau en utilisant la boucle *foreach* la plus adaptée.

Définissez la fonction `traduitEnAnglais` qui prend en paramètre le nom d'un jour de la semaine en français ainsi que le tableau des jours de la semaine créé dans cette question. Cette fonction retournera le nom du jour de la semaine en anglais lorsque le jour existe, la chaine de caractères "*Unknown*" sinon. Testez la fonction `traduitEnAnglais` .

Définissez la fonction `traduitEnFrancais` qui prend en paramètre le nom d'un jour de la semaine en anglais ainsi que le tableau des jours de la semaine créé dans cette question. Cette fonction retournera le nom du jour de la semaine en français lorsque le jour existe, la chaine de caractères "*Inconnu*" sinon. Testez la fonction `traduitEnFrancais` .