Le langage PHP

02 -Variables et Types

## Question 1 - Comprendre un programme

### Partie 1

$number1 = 10;

*Déclaration d’une variable nommée* ***$number1*** *et affectation de la valeur* ***10*** *à cette variable.*

* *typage implicite (****nombre entier****)*

$number2 = 4;

*Déclaration d’une variable nommée* ***$number2*** *et affectation de la valeur 4 à cette variable.*

* *Typage implicite (****nombre entier****)*

$message = 'Le résultat de la division vaut : ';

*Déclaration d’une variable nommée* ***$message*** *et affectation de la valeur ‘Le résultat de la division vaut : ’ à cette variable.*

* *Typage implicite (****chaîne de caractères****)*

$result = $number1 / $number2

*Déclaration d’une variable nommée* ***$result*** *et affectation de la valeur du résultat de la division entre les variables* ***$number1*** *et* ***$number2*** *à cette variable.*

* *Typage implicite (****float****)*

echo $message . $result . '<br>';

*Affichage dans le navigateur de la concaténation des variables* ***$message*** *et* ***$result*** *suivis d’une chaine de caractères ayant pour valeur ‘<br>’.*

### Partie 2

$isTrue = true;

*Déclaration d’une variable nommée* ***$isTrue*** *et affectation de la valeur* ***true*** *à cette variable.*

* *typage implicite (****booléen****)*

$isFalse = !$isTrue;

*Déclaration d’une variable nommée* ***$isFalse*** *et affectation de la valeur inverse à la variable* ***$isTrue*** *à cette variable.*

* *typage implicite (****booléens****)*

$message = 'La variable booléenne "isFalse" vaut : ';

*Déclaration d’une variable nommée* ***$message*** *et affectation de la valeur ‘La variable booléenne «isFalse» vaut : ’ à cette variable.*

* *Typage implicite (****chaîne de caractères****)*

echo $message . ($isFalse ? 'true' : 'false') . '<br>';

*Affichage dans le navigateur de la concaténation des variables* ***$message*** *et d’une opération conditionnel ternaire avec la variable* ***$isFalse*** *(si celle-ci à pour valeur true, alors il affichera la chaine de caractères ‘true’, sinon il affichera la chaine de caractères ‘false’) suivis d’une chaine de caractères ayant pour valeur ‘<br>’.*

### Partie 3

$number3 = 4.56713;

*Déclaration d’une variable nommée* ***$number3*** *et affectation de la valeur* ***4.56713*** *à cette variable.*

* *typage implicite (****float****)*

$otherResult = $number3 \* $number2;

*Déclaration d’une variable nommée* ***$otherResult*** *et affectation de la valeur du résultat de la multiplication entre les variables* ***$number3*** *et* ***$number2*** *à cette variable.*

* *Typage implicite (****float****)*

$message = 'Le résultat de la multiplication vaut : ' . $otherResult;

*Déclaration d’une variable nommée* ***$message*** *et affectation de la valeur ‘Le résultat de la multiplication vaut : ’ suivis de la concaténation de la variable* ***$otherResult*** *à cette variable.*

* *Typage implicite (****chaîne de caractères****)*

echo $message;

*Affichage dans le navigateur de la variable* ***$message****.*

echo '<br>';

*Affichage dans le navigateur de la chaine de caractères ‘<br>’. Ici elle sera prise comme une balise HTML et ne sera donc pas afficher directement mais traiter en HTML par le navigateur.*

## Question 2 - Réflexions

### Partie 1 - Client et serveur

*Depuis l’inspecteur de Chrome, j’observe une suite de chaines caractères contenant les résultats des différentes opérations de mon script suivis de balises HTML <br>.*

### Partie 2 - PHP et HTML

*La fonction php echo permet d’afficher la valeur d’une variable dans le navigateur. Si l’on ajoute à la suite dans le script PHP plusieurs fonction echo, les valeurs de celle-ci se succèderont sans espace et retour à la ligne. La balise HTML <br> et simplement là pour faire un retour à la ligne après avoir affiché la valeur d’une variable avec la fonction echo.*

### Partie 3 - Opérateur ternaire

*L’opérateur ternaire « ? : » fonctionne comme l’opérateur « if else ». Avant l’opérateur « ? », on doit lui assigner une condition. Dans cette exemple la condition est : « est ce que la variable* ***$isFalse*** *à pour valeur* ***true****». Une fois la condition posée, on doit lui assigner une valeur pour si la condition est* ***vraie*** *(true) et une autre valeur pour si la condition est* ***fausse*** *(false). Entre ces deux valeurs, on ajoute l’opérateur « : » pour les séparer. Dans cet exemple, si la condition est vraie, alors le résultat de l’opération sera la chaine de caractères « true », et si la condition est fausse, alors le résultat de l’opération sera la chaine de caractères « fause ».*