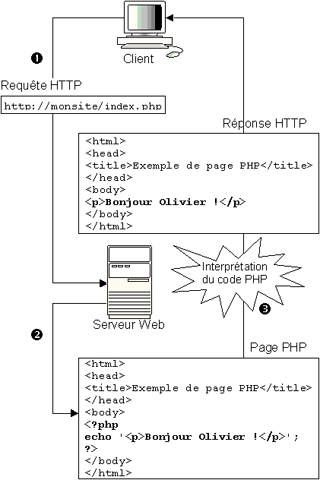
**Schéma synthétisant le traitement d’un fichier PHP par le serveur Web** : tiré de PHP7 – ENI – Olivier Heurtel



Deux principes pour mixer du code HTML et du code PHP :

* La page peut contenir une ou plusieurs inclusions de code PHP.
* Le code PHP génère du "texte" qui est intégré dans la page HTML envoyée au navigateur. Tout "texte" compréhensible par le navigateur peut donc être généré par le code PHP : du texte simple, du code HTML, du code JavaScript...

**Déclaration de code PHP**

**Ouverture**

<?php

**Fermeture**

?>

D’autres syntaxes existent, non recommandées.

**Fonction de base**

La fonction echo est la fonction de base de toute page PHP. Elle permet d’afficher une ou plusieurs chaînes et donc d’inclure du texte dans la page HTML envoyée au navigateur.

Syntaxe

echo(chaîne texte)

echo chaîne texte[,...]

texte : texte à afficher.

La première syntaxe n’accepte qu’un paramètre alors que la deuxième en accepte plusieurs.

Exemple : cf fonction\_base\_echo.php

**Séparation des instructions**

En PHP, toutes les instructions doivent se terminer par un point-virgule.

**Commentaires**

PHP propose 3 syntaxes :

1. // ou # pour insérer du commentaire sur une ligne "dédiée" ou à la suite d’une instruction.
2. /\* ... \*/ pour insérer du commentaire sur plusieurs lignes.
3. PHPDoc : pour générer de la documentation automatiquement :
   1. /\*\* pour démarrer
   2. Chaque ligne commence par une \*
   3. \*/ pour terminer

**Définition et portée d’une constante**

La portée d’une constante est le script dans lequel elle est définie. En général par convention les constantes sont en majuscule et en début de script.

**Syntaxe :**

define(‘nom’,’valeur’);

**Variables – Définition, portée, durée**

Typage automatique. En version 7, possibilité d’utiliser un typage strict.

Les noms de variables sont précédés d’un $ et sensibles à la casse.

La portée d’une variable est le script dans lequel elle est définie. La durée de vie d’une variable est le temps de l’exécution du script. Lorsque le script se termine, les variables sont supprimées.

Affectation par valeur d’une variable à une autre variable:

$cible = $source ;

La valeur de $source est copiée pour initialiser la variable $cible. Si $source est modifié ensuite, cela ne change rien à la valeur de $cible.

Affectation par référence d’une variable à une autre variable (notion similaire aux pointeurs) :

$cible = &$source ;

$source est passée par référence à la variable $cible. Si $source est modifié ensuite, cela peut changer la valeur de $cible (selon l’ordre des instructions).

## **Tableaux – Définition – Utilisation – Parcours d’un tableau**

En PHP, un tableau est une collection (liste d’éléments) ordonnée de couples clé/valeur.

La clé peut être

1. type entier : le tableau est dit numérique et la clé est désignée par le terme indice.
2. type chaîne : le tableau est dit associatif : les clés ne sont pas forcément consécutives, ni ordonnées, et ce tableau peut présenter des clés entières et des clés de type chaîne.

La valeur associée à la clé peut être de n’importe quel type, et notamment de type tableau ; dans ce cas, le tableau est dit multidimensionnel.

La fonction array permet de créer un tableau à partir d’une liste d’éléments.

**Syntaxe**

array([mixte valeur[, ...]])

ou

array([{chaîne | entier} clé => mixte valeur[, ...]])

|  |  |
| --- | --- |
| valeur | Élément du tableau. |
| clé | Valeur de la clé. |

3 principales méthodes pour parcourir un tableau

1. contrôle itératif avec **for**
2. contrôle itératif avec boucle **while**
3. parcours de tableau avec **foreach**

La 3ème est la plus simple et celle que nous utiliserons majoritairement.

**Foreach : récupération de valeurs dans une variable**

Syntaxe

foreach(tableau as variable)

**Foreach : récupération de paires clef / valeur sous forme de variables**

la structure foreach permet de parcourir un tableau de tableaux et de récupérer les éléments du tableau imbriqué dans des variables à l’aide de la fonction list.

Syntaxe

foreach(tableau as list(variable[,... ]))

**Les fonctions**

Une fonction peut être définie comme un sous programme appelé depuis le programme principal.

En PHP, le terme fonction recouvre la notion de

* Procédure : exécute un ensemble d’instructions mais ne retourne rien
* Fonction : exécute un ensemble d’instructions et retourne quelque chose : une valeur, une variable

**Fonctions PHP**

PHP propose un certain nombre de fonctions utiles sur les **variables** :

| **Nom** | **Syntaxe** | **Rôle** | **Retourne** |
| --- | --- | --- | --- |
| empty | booléen empty(variable) | Indique si une variable est vide ou non. | boolean |
| isset | booléen isset(variable) | Indique si une ou plusieurs variables sont définies ou non. | boolean |
| unset | unset(variable) | Supprime une ou plusieurs variables. |  |
| var\_dump | var\_dump(variable) | Affiche des informations sur une ou plusieurs variables (type et valeur). Utilisé en phase de développement |  |

Plusieurs centaines de fonctions PHP existent, dans tous les domaines : manipulation de chaînes, nombres, réseau, web etc

Principales fonctions sur les **tableaux** :

Les fonctions les plus utilisées sont les suivantes :

| Nom | Rôle |
| --- | --- |
| count | Compte le nombre d’éléments dans un tableau. |
| in\_array | Teste si une valeur est présente dans un tableau. |
| array\_search | Recherche une valeur dans un tableau. |
| array\_replace | Remplace des valeurs dans un tableau. |
| [a|k][r]sort | Trie un tableau (plusieurs variantes possibles). |
| explode | Découpe une chaîne selon un séparateur et stocke les éléments dans un tableau. |
| implode | Regroupe les éléments d’un tableau dans une chaîne à l’aide d’un séparateur. |
| max | Retourne la plus grande valeur stockée dans un tableau. |
| min | Retourne la plus petite valeur stockée dans un tableau. |
| str\_split | Découpe une chaîne en morceaux de longueur fixe et stocke les éléments dans un tableau. |
| array\_column | Retourne les valeurs d’une colonne d’un tableau multidimensionnel. |

**Fonctions de date**

Principales fonctions de date :

Plusieurs fonctions permettent de manipuler les dates sous l’une ou l’autre de ces formes :

| Nom | Rôle |
| --- | --- |
| checkdate | Vérifie que trois entiers représentant le jour, le mois et l’année correspondent à une date valide. |
| date | Convertit en chaîne une date donnée sous la forme d’un timestamp Unix. |
| getdate | Stocke dans un tableau les différentes composantes d’une date donnée sous la forme d’un timestamp Unix. |
| date\_parse\_from\_format | Stocke dans un tableau les différentes composantes d’une date donnée sous la forme d’une chaîne de caractères. Ajoutée en version 5.3. |
| time | Donne le timestamp Unix actuel. |

**Fonctions manipulation de fichier**

Les fonctions les plus courantes sont les suivantes :

| Nom | Rôle |
| --- | --- |
| fopen | Ouvrir un fichier |
| fclose | Fermer un fichier |
| fread | Lire le contenu d’un fichier (dans une chaîne) |
| file | Lire le contenu d’un fichier (dans un tableau) |
| readfile | Lire le contenu d’un fichier et l’envoyer directement vers la sortie |
| fwrite | Écrire dans un fichier |
| file\_get\_contents | Ouvrir, lire et fermer un fichier |
| file\_put\_contents | Ouvrir, écrire et fermer dans un fichier |
| copy | Copier un fichier |
| unlink | Supprimer un fichier |
| rename | Renommer un fichier |
| file\_exists | Tester l’existence d’un fichier |
| filesize | Lire la taille d’un fichier |
| chdir | Changer de répertoire courant |
| opendir | Ouvrir un répertoire |
| closedir | Fermer un répertoire |
| readdir | Lire le contenu d’un répertoire |
| scandir | Liste le contenu d’un répertoire (dans un tableau) |

**Fonctions utilisateur**

Ce sont des fonctions définies par le développeur.

Syntaxe :

Function nom\_fonction($argument1, …)

{

// Instructions

// Mot clef return si la fonction renvoie une valeur

}

Pour retourner une valeur, le mot clef ***return*** est utilisé à l’intérieur du bloc d’instructions.

Nombre de paramètres variable : on utilise l’opérateur de jonction, composé de 3 points : …

Exemple : liste\_arguments\_fonction.php

**Inclusion d’un fichier**

**Utilisation**

2 grands types d’utilisation :

* Inclure des définitions statiques : constantes, définitions de fonctions.
* Inclure du code PHP ou HTML dynamique qui s’exécute effectivement au moment de l’inclusion : section HTML commune à plusieurs pages (en-tête, pied de page)

Syntaxe

include(fichier)

include\_once(fichier)

Include once permet de s’assurer que le fichier n’est inclus qu’une seule fois.