

SUPPORT DE COURS Bloc 1 PHP



PHP

date	révision
	David Roumanet : création du document
Janvier 2019	David Roumanet : Amélioration (v.83
Décembre 2019	Modifications diverses
BTS SI	O BTS SIO (SLAM)
@09	Bloc1 - PHP

BTS SIO	BTS SIO (SLAM)
@ ① ③ ② BY NC SA	Bloc1 - PHP

1. INTRODUCTION

Langage créé en 1994 par Rasmus Lerdorf pour créer son site Web (on est au tout début du protocole HTTP et du langage HTML).

Rapidement, signifie : *PHP: Hypertext Preprocessor*. Devient le langage de programmation Web **côté serveur** (contrairement à JavaScript par exemple) le plus utilisé.

Enfin, PHP est un langage gratuit et dont le source est disponible en libre accès.

Il y a eu de nombreuses versions de PHP mais actuellement, deux versions cohabitent :

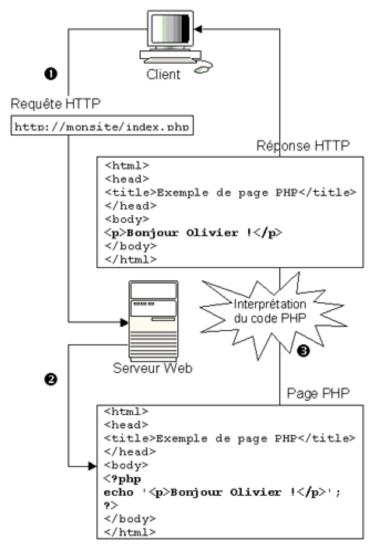
- La version 7 qui est la plus moderne et la plus performante, mais aussi la plus incompatible avec les anciens scripts (sortie en décembre 2015), actuellement en version 7.4
- La version 5.6 qui correspond à la dernière version de la génération 5 (6 n'est pas sorti, par superstition ?)

2. FONCTIONNEMENT

PHP peut fonctionner comme langage de script (au même titre que PowerShell, Ruby, Perl).

Toutefois, le fonctionnement des sites web est un peu particulier et c'est le serveur web qui appelle les scripts. Voici le schéma simplifié de fonctionnement :

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



Etape 1: demande par le client d'une page PHP

<u>Etape 2</u>: le serveur Web transmet les parties PHP au moteur PHP qui exécute les morceaux correspondant. Le résultat est renvoyé au navigateur.

Schéma détaillé du fonctionnement :

- Le client ouvre une connexion sur le serveur web, sur le port 80 par défaut.
- Le serveur transfère une page par défaut (généralement index.php ou index.html)
- Le client renvoie une demande de page (et éventuellement joint des données)
- Le serveur exécute la requête et le script PHP correspondant. En général, le script créé une page au format HTML.

Web browser at client side

HTML pages

Apache web server

Apache PHP pages

Cet aspect implique que pour tester du code

PHP, il faut avoir un *serveur web actif* : cela peut être Microsoft IIS, Apache de l'Apache Software Foundation, NGINX d'Igor Sysoev ou d'autres serveurs.



Tableau 2.1: les serveurs web les plus connus

BTS SIO	BTS SIO (SLAM)
© (1) SA	Bloc1 - PHP

Pour faciliter <u>l'installation</u> d'un ensemble de fonctions et de services, il existe plusieurs solutions gratuites

gratuites	
EasyPHP	EasyPHP intègre les services Apache, PHP et MySQL. Il propose deux versions : Devserver et Webserver (censé permettre la mise en production de site). Cette version est complètement configurable, et permet d'arrêter ou démarrer facilement les services (web ou base de données) et est portable. Taille modeste
WAMP	WAMP est similaire, avec les services Apache, PHP et MySQL. Taille importante
UwAmp	UwAmp intègre lui aussi Apache, PHP et MySQL. Il propose des raccourcis graphiques (pratique pour les débutants) et est portable (peut fonctionner depuis une clé USB, sans installation préalable) Taille très modeste
XAMPP	XAMPP se veut totalement libre, et ne propose que des logiciels libres et opensource. Apache, MariaDB, PHP et Perl. Taille moyenne.
Winginx	Winginx utilise NGINX comme moteur web. Pour le reste, on retrouve PHP et MySQL (mais aussi NodeJS, MongoDB, Redis). Taille modeste
Laragon	Mon chouchou: laragon, https://laragon.org/ permet également de générer les squelettes des principaux framework PHP (Laravel, Symfony) Taille importante

Tableau 2.2: choisissez votre solution

On peut également écrire et exécuter directement du code PHP en ligne, dans un Sandbox (bac à sable).

Par exemple:

https://wtools.io/php-sandbox

https://3v4l.org/

https://repl.it/languages/php

PHP et MySQL ne sont pas les seuls langages pour créer et animer des sites web : **ASP.net** de Microsoft, **Ruby on Rails, Django** (Python), **Java EE** (JSP framework Spring et SpringBoot) permettent de rendre un site web dynamique. De même **SQL server** de Microsoft, **MariaDB** (clone de **MySQL**), **Oracle** et **PostGreSQL** sont d'autres bases de données reconnues.

BTS SIO	BTS SIO (SLAM)
@ ① ③ ② BY NC SA	Bloc1 - PHP

3. LA SYNTAXE PHP

PHP étant devenu structuré et orienté objet, de nombreux concepts sont déjà connus.

3.1 HTML ET PHP

Le lien entre PHP et HTML est très fort. Dès lors, les scripts PHP peuvent se placer dans n'importe quelle partie d'un fichier HTML, du moment que l'on utilise les balises.

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...

3.1.1 Déclaration de code PHP

3.1.1.1 Ouverture de code

<?php /* Code PHP */</pre>

3.1.1.2 Fermeture de code¹

?>

3.1.1.3 Code PHP

Mais surtout, cette zone devient un espace disponible pour PHP : un affichage peut se faire à l'endroit même avec l'instruction echo.

```
<?php echo "Le SIO c'est pas toujours \"rigolo\""; ?>
```

Quel que soit l'endroit entre les balises <body> et </body> où se trouve ce script, s'affichera alors Hello "world".

3.1.2 Extension de fichier et ouverture

Maintenant, il reste préférable de choisir l'extension .php pour les fichiers PHP.

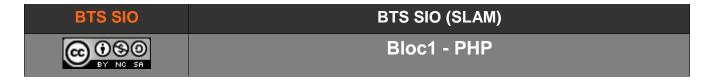


Dernière chose importante : l'URL dans le navigateur doit impérativement comporter l'adresse IP (ou le nom DNS), et éventuellement le port, du serveur web. En effet, si le navigateur peut afficher le fichier, il ne peut pas exécuter les scripts PHP.

Exemple d'erreur classique : file:///E:/uWamp_3/www/monFichierHTML.html

BTS SIO ENC 2021 6

¹ D'autres syntaxes existent, non recommandées



3.2SYNTAXE DE BASE

Comme il s'agit d'un langage scripté, les instructions sont a écrire entre les balises <?php et ?>

3.2.1 Commentaires

Les commentaires sur une ligne utilisent un // tandis que commenter un paragraphe se fait avec /* et */

3.2.2 Instructions

On trouve le symbole point-virgule ';' à la fin des instructions, et les crochets pour les blocs { et }.

En revanche, PHP conserve certains défauts d'origine pour ceux qui viennent des langages typés comme C#, C++ ou Java...

3.2.3 Génération de documentation PHPDoc

PHPDoc : pour générer de la documentation automatiquement :

- 1. /** pour démarrer
- 2. Chaque ligne commence par une *
- 3. */ pour terminer

3.2.4 Nommage des variables

Ainsi, la déclaration d'une variable ne nécessite pas par défaut de type sur les variables, qui commencent par un \$.2

```
$\text{sphp}
$nom = "Salut les SIO";

// Ceci est un commentaire

function ecrire($d)
{
    return "-".$d."-";
}
echo ecrire($nom);
```

Vous noterez que la concaténation de chaîne ne se fait plus avec le symbole + mais avec le point '.' car le fait de ne pas avoir de type sur les variables implique que l'addition soit numérique uniquement.

BTS SIO ENC 2021 7

² A partir de la version 7, PHP permet une option qui donne la possibilité d'imposer un typage strict des variables et fonctions PHP: declare(strict_types = 1);

BTS SIO	BTS SIO (SLAM)
© (1) SO BY NC SA	Bloc1 - PHP

Extrait doc PHP:

Un nom de variable valide doit commencer par une lettre ou un souligné (_), suivi de lettres, chiffres ou soulignés. Exprimé sous la forme d'une expression régulière, cela donne : $^[a-zA-z]\times 0-xff[[a-zA-z]-9]\times 0-xff]*$

3.2.5 Contenu d'une variable

3.2.5.1 Affectation de contenu à une variable

```
$nom="Robert";
echo "Le nom est: ".$nom ."<BR>";
$nom = 45;
echo "L'âge est: ".$nom;
```

Comme PHP n'est pas typé par défaut, on peut mettre un peu n'importe quoi dans une variable ... Ici la variable \$nom contient d'abord une chaîne de caractères puis ensuite un entier.

3.2.5.2 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.

```
$source=45;
echo "L'age est: ".$source ."<BR>";
$cible = $source;
$source = 50;
echo "L'ancien âge est: ".$cible;
}
```

3.2.5.3 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).

A éviter. Il faut savoir que ça existe, vous pouvez en trouver dans du code « legacy ». (une question làdessus en TD).

BTS SIO	BTS SIO (SLAM)
© (1) SO BY NC SA	Bloc1 - PHP

Exemple : dans le script précédent, quel est le résultat si on change \$cible = \$source ; par
\$cible = &\$source ; 2

3.2.1 Portée d'une variable

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.

L'endroit dans le script où on déclare une variable va déterminer l'endroit où la variable va être accessible c'est-à-dire utilisable.

Pour simplifier, une variable peut être :

- globale : comportement par défaut
- locale : si elle est définie dans une fonction

3.3TABLEAUX - DEFINITION ET UTILISATION

3.3.1 Définition d'un tableau

En PHP, les tableaux sont des types de données structurés permettant de grouper des informations ensemble.

Les données sont stockées sous forme de collection (liste d'éléments) ordonnée de couples clé/valeur.

La clé peut être

- 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.

3.3.2 Exemple de tableau avec index numérique

```
// Déclaration d'un tableau indexé numériquement
$matieres = array('SI1','SI2','SI3','SI4');
```

L'élément \$matieres[0] contient la valeur SI1 et ainsi de suite jusqu'à \$matieres[3] qui contient la valeur SI4.

3.3.3 Exemple de tableau associatif

```
// Déclaration d'un tableau associatif

$identite = array(
    'nom' => 'Robert',
    'prenom' => 'Timothée',
    'age' => 45,
    'estEtudiant' => false

a);
```

Chaque élément du tableau est un couple (clef, valeur).

L'élément \$identite['prenom'] renvoie Timothée, l'élément \$identite['age'] renvoie 45 etc

3.3.4 Ajout d'éléments dans un tableau

3.3.4.1 Pour un tableau indexé numériquement :

Il suffit d'ajouter l'élément dans le tableau.

```
$matieres[] = 'SLAM5';
```

Ajoute la valeur SLAM5 à la fin du tableau \$matieres.

3.3.4.2 Pour un tableau associatif:

Entre les crochets [] on indique la clef et à droite du signe = on affecte la valeur correspondante pour former un nouveau couple (clef, valeur).

```
// Ajout de la taille de la personne dans le tableau associatif
$identite['taille'] = 180;
```

3.4 LES FONCTIONS

Elles s'écrivent avec des parenthèses à la fin. La richesse de PHP est d'avoir de très (très) nombreuses fonctions prédéfinies et de pouvoir également en définir de nouvelles personnalisées.

3.4.1 Définition

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
- <u>Fonction</u> : exécute un ensemble d'instructions et retourne quelque chose : une valeur, une variable



3.4.2 Fonction et portée des variables

Par défaut, Toute variable utilisée dans une fonction utilisateur est, par définition, locale.

Pour utiliser une variable globale dans une fonction, il faut la définir avec le mot clef global devant.

3.4.3 appeler une fonction : date

Dans l'exemple suivant, nous utilisons une fonction déjà existante : date()

```
Aujourd'hui nous sommes le <?php echo date('d/m/Y h:i:s'); ?>
```

affichera

Aujourd'hui nous somme le 28/01/2018 23:30:00

3.4.4 Créer une fonction utilisateur

Pour créer sa propre fonction en PHP, il suffit d'encadrer le code entre deux accolades et de préciser le nom de la fonction. Il n'est pas nécessaire d'indiquer le type de retour (dans la signature) :

```
function monMax($a, $b) {
// le code qui définit et calcule $max
return $max;
}
```

On distingue de manière générale dans la plupart des langages de programmation 2 types de fonction (souvent appelées méthode ou opération) : les *fonctions*, qui renvoient quelque chose et les *procédures*, qui effectuent un traitement mais ne renvoient rien.

```
Fonction PHP de type procédure : ne renvoie rien Fonction PHP de type fonction : renvoie qq chose

function jeSaisTout($matiere){
    echo "je suis un spécialiste de: ".$matiere ;
}

jeSaisTout( matiere: "Java");

echo jeSaisPeu( matiere: "Haskell") ;
```

BTS SIO ENC 2021 11

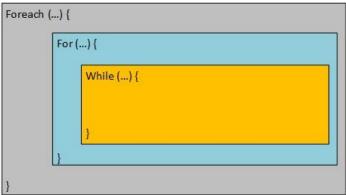
4. LES BOUCLES ET LES CONDITIONS EN PHP

4.1 LES BOUCLES

Comme les autres langages étudiés, PHP dispose de boucles (répétitions) et de conditions.

4.1.1 Introduction

La notion de répétition avec sortie prédéfinie ou conditionnelle est implémentée en PHP. Les instructions contenues dans le bloc seront répétées un certain nombre de fois. Il est bien sûr possible d'imbriquer les conditions.



4.1.2 La boucle For

La boucle de comptage standard : on initialise la variable, on donne la condition de sortie et enfin on précise le pas d'incrémentation ou de décrémentation :

```
for ($t = 0; $t < 10; $t++) {
    echo 'La variable $t vaut '.$t.'<br>';
```

4.1.3 while et do... while

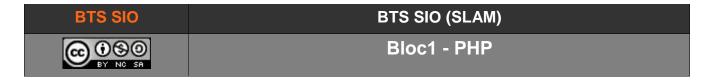
Il s'agit de deux boucles dont la sortie se fera lorsque la condition sera fausse. Dans le premier cas, le test est fait au début de la boucle, ce qui signifie que le bloc peut n'être jamais exécuté.

Dans le deuxième cas, on exécute les instructions du bloc et on teste la condition à la fin. Comparez

```
$coffre = 0; $ration = 2;

while ($coffre < 10) {
    echo "le coffre n'est pas encore plein... <br>";
    $coffre = $coffre + $ration;

}
echo "le coffre est maintenant a $coffre<br>";
```



et

```
$coffre = 0; $ration = 2;
do {
    echo "le coffre n'est pas encore plein... <br>";
    $coffre = $coffre + $ration;
} while ($coffre < 10);
echo "le coffre est maintenant a $coffre < br > ";
```

Dans le 2^{ème} cas le bloc d'instructions est donc exécuté au moins une fois.

4.1.1.3 Foreach : ma boucle préférée

Similaire dans le fonctionnement interne, c'est PHP qui détermine le nombre d'élément d'un tableau ou d'une liste à parcourir. Le programmeur sait seulement que la liste sera parcourue dans l'ordre.

Parcours tableau indexé avec boucle foreach

```
$arr = array(1, 5, 3, 4, 5, 1, 2, 2, 3, 4);

foreach ($arr as $value) {
    echo "La note est de: $value /5 soit encore ".$value*4 ." / 20 <BR>";

}
```

Ici, le programme affiche la table des notes d'une classe, multipliée par 4 pour afficher une note sur 20.

C'est la boucle la plus utilisée car la plus simple à manipuler et la plus naturelle.

Parcours tableau associatif avec boucle foreach

```
$capitales = array('GB' => 'Londres', 'Allemagne' => 'Berlin', 'France' => 'Paris', 'Italie' => 'Rome');

foreach ($capitales as $unPays => $uneCapitale)
{
    echo "La capitale du pays: ".$unPays. " est: ".$uneCapitale. "<BR>";
}
```

Ici le tableau associatif associe chaque pays avec sa capitale. Dans la boucle foreach, on utilise la syntaxe :

```
foreach ($tableau as $clef => $valeur) {
    // traitements
}
```



On peut également faire des choses plus avancées : 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**.

4.1.4 Le branchement conditionnel et les conditions

4.1.4.1 Introduction

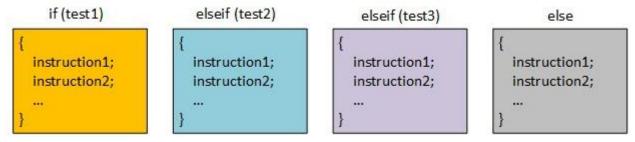
Les conditions permettent de faire des tests, de contrôler des hypothèses.

Il existe un ensemble d'instructions : if, elseif et else qui vont évaluer des expressions, c'est ce qu'on appelle une condition et en fonction du résultat, vrai ou faux, exécuter le bloc d'instructions correspondant.

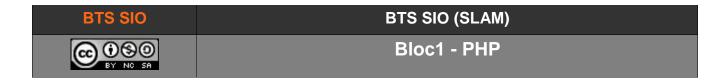
Mais il y a aussi un ensemble pour les décisions nombreuses : switch, case, et default.

4.1.4.2 If elseif et else

Ces instructions permettent de créer des blocs qui ne seront exécutés que si la condition est vraie.



Le premier test utilise le mot-clé 'if' tandis que les suivants utilisent 'elseif'. Enfin, 'else' permet d'exécuter un dernier bloc si les autres blocs n'ont pas été exécutés.



En code PHP, cela donne les instructions suivantes :

```
lif ($role == "administrateur") {
    echo "connecté en tant qu'administrateur<br>";
l} elseif ($role == "rédacteur") {
    echo "connecté en tant que rédacteur<br>";
l} else {
    echo "vous êtes un utilisateur<br>";
}
```

4.1.4.3 Opérateur de comparaison

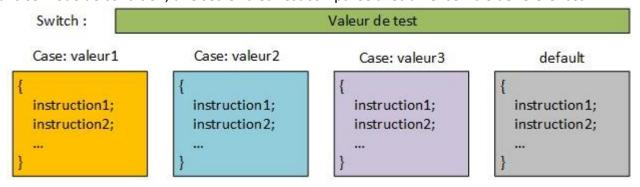
L'opérateur de comparaison en PHP est == à ne pas confondre avec l'opérateur d'affectation = .

```
if ($nom = "robert")
{
    echo "drôlement chouette comme nom".$nom;
}
```

Le code ci-dessus affichera tout le temps l'instruction echo! car l'expression \$nom = "robert" sera toujours vraie et réalisera l'affectation de la valeur robert à la variable \$nom.

4.1.4.4 switch, case et default

Dans ce mode de condition, une seule valeur est comparée avec un ensemble de références.



En code PHP, cela donne les instructions suivantes :



```
switch ($avisFiabilite) {
    case 0:
        echo "0/5 : ce produit est inutilisable !";
        break;
    case 1:
        echo "1/5 : ce produit n'est pas fiable !";
        break;
    case 2:
        echo "2/5 : ce produit fonctionne parfois !";
        break;
    case 3:
        echo "3/5 : le produit est moyen !";
        break;
    default:
        echo "> 3/5 : un produit correct !";
        break;
}
```

4.1.4.5 l'opérateur ternaire : mon choucou

C'est l'opérateur ?:

Extrait de la documentation PHP:

L'expression (expr1) ? (expr2) : (expr3) est évaluée à expr2 si expr1 est évaluée à **true**, et expr3 si expr1 est évaluée à **false**.

<u>Exemple</u>: soit l'algorithme suivant en PHP. Si ma note de SI6 est supérieure ou égale à 10, j'aurai du chocolat, sinon je mangerai du navet.

L'opérateur ternaire est surpuissant et largement utilisé dans un grand nombre de langages de programmation.

BTS SIO	BTS SIO (SLAM)
© (1) SA	Bloc1 - PHP

5. ANNEXE - Fonctions PHP

5.1 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
IIICCAT I		Indique si une ou plusieurs variables sont définies ou non.	boolean
unset	unset(variable)	Supprime une ou plusieurs variables.	
var_dump	,	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

5.2 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.	

BTS SIO	BTS SIO (SLAM)
© (1) SA	Bloc1 - PHP

5.3 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.
lidate	Convertit en chaîne une date donnée sous la forme d'un timestamp Unix.
	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.

5.4 Fonctions de 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
сору	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

BTS SIO	BTS SIO (SLAM)
⊕ ⊕ ⊕ ⊕ ⊕ ⊕ ⊕ ⊕ ⊕ ⊕	Bloc1 - PHP

readdir	Lire le contenu d'un répertoire
scandir	Liste le contenu d'un répertoire (dans un tableau)