

Instructions



Instructions	Commentaires (1)	PHP 2
<p align="center">Commentaire simple : // ou #</p>		
<pre>// Initialisation des variables de script \$nom_etudiant = 'Pierre'; \$notes = array(10, 14, 'abs', 12, 'abs'); \$moyenne = false;</pre>		
<pre>define('DEBUG', TRUE); // Affichage détaillé des erreurs define('APPLI', 'eBlog'); // Sert pour nom cookie et répertoire</pre>		
<pre>if (\$Password == '') { appli_exit(-4); //===>>> EXIT : ERREUR mot de passe }</pre>		
<pre>// Initialisation totaux étudiant \$tots = array(0, // Numéro étudiant '', // Nom et prénom 0, // Durée formation en heures array(), // Notes obtenues aux modules 0); // Nombre d'heures présence</pre>		
<p align="center"><small>francois.piat@univ-fcomte.fr</small></p>		

Instructions

Commentaires (1)

PHP
3

```
// L'art de mettre des commentaires inutiles

$nom_etudiant = 'Pierre';           // Nom de l'étudiant
$notes = array(10, 14, 'abs', 12, 'abs'); // tableau des notes
$moyenne = false;                   // Moyenne des notes

if ($cours == 'PHP') {              // Si cours PHP la salle est amphi B
    $salle = 'Amphi B';
}
```

francois.piat@univ-fcomte.fr

Instructions

Commentaires (2)

PHP
4

Commentaire multilignes : /* */

```
/*
Traitement ajout/modif d'une page de mise à jour.
On vérifie si les zones saisies sont valides.
Si oui on met à jour la BD puis on génère la page de fin.
*/
if ($etat == STATE_END_ADD
|| $etat == STATE_END_UPD)
{
    if (form_verify()) {
        update_bd_produits();
        update_bd_ventes();
        page_fin();
        return;           //====>>> RETURN
    }
}
```

francois.piat@univ-fcomte.fr

Instructions

Commentaires (3)

PHP 5

Commentaire multilignes : `/*` `*/`

```
/**
 * Vérification d'un paramètre GET
 *
 * Vérification que des paramètres passés dans un GET (url)
 * sont numériques et d'une valeur minimum.
 * En cas d'erreur, la fonction retourne FALSE
 *
 * @param string $nom Nom du paramètre à vérifier
 * @param integer $mini Valeur minimum du paramètre
 * @param boolean $requis Paramètre obligatoire ou pas
 * @global array $_GET
 *
 * @return integer Valeur du paramètre ou FALSE
 */
function verifGET($nom, $mini = 0, $requis = TRUE) {
    ...
}
```

www.phpdoc.org
francois.piat@univ-fcomte.fr

Instructions

echo et print

PHP 6

- Afficher une chaîne = envoyer la chaîne au navigateur
- Instruction **echo**

```
echo '<h1>', $titre, '</h1><p>', $txt, '</p>';
```
- Instruction **print**

```
print '<h1>'.$titre.'</h1><p>'.$txt.'</p>';
```
- Préférer l'instruction **echo**

francois.piat@univ-fcomte.fr

Instructions

if

PHP 7

```
if (expression) {
    instruction(s)
}
```

Conseil : toujours délimiter les blocs d'instructions avec des accolades {}

```
if ($cours == 'PHP') {
    $salle = 'Amphi B';
    $prof = 'François Piat';
    afficheHeures('PHP');
}
```

```
if ($note >= 10) {
    validerModule();
}
```

```
($note >= 10) && validerModule();
```

francois.piat@univ-fcomte.fr

Instructions

if else

PHP 8

```
if (expression) {
    instruction(s)
} else {
    instruction(s)
}
```

Conseil : toujours délimiter les blocs d'instructions avec des accolades {}

```
if ($type == 'cours') {
    $salle = 'Amphi B';
    $prof = 'François';
} else {
    $salle = '317C';
    $prof = 'Louis';
}
```

```
if ($nbAbsences == 0) {
    $absent = false;
} else {
    $absent = true;
}
```

```
$absent = ($nbAbsences > 0);
```

```
if ($note < 10) {
    vaSession2();
} else {
    calculMention($note);
}
```

```
($note < 10) ? vaSession2() : calculMention($note);
```

francois.piat@univ-fcomte.fr

Instructions
if elseif
PHP 9

```

if (expression) {
    instruction(s)
} elseif (expression) {
    instruction(s)
}

```

**Conseil : toujours délimiter
les blocs d'instructions
avec des accolades {}**

```

if ($note >= 16) {
    $mention = 'Très bien';
} elseif ($note >= 14) {
    $mention = 'Bien';
} elseif ($note >= 12) {
    $mention = 'Assez bien';
} else {
    $mention = 'Passable';
}

```

```

if ($nom == 'Piat') {
    $salle = 'Amphi C';
} elseif ($groupe != 'A') {
    $salle = 'Amphi C';
} elseif ($mois < 3) {
    $salle = 'Amphi C';
} else {
    $salle = 'Amphi A';
}

```

francois.piat@univ-fcomte.fr

Instructions
switch
PHP 10

```

switch (expression) {
case valeur1:
    instruction(s);
    break;
case valeur2:
    instruction(s);
    break;
case valeur3:
case valeur4:
case valeur5:
    instruction(s);
    break;
default:
    instruction(s);
}

```

```

switch ($cours) {
case 'PHP':
    $salle = 'Amphi B';
    break;
case 125:
    $salle = 'Amphi D';
    break;
case 'HTML':
case 'JavaScript':
    $salle = 401;
    break;
case FALSE:
    $salle = 'Cours annulé';
    break;
default:
    $salle = FALSE;
}

```

francois.piat@univ-fcomte.fr

Instructions
for (1)
PHP 11

Initialisations
Test d'itération
Compteurs

```

for (expression1; expression2; expression3) {
    instruction(s)
}

```

```

$nb = 5;
for ($i = 0; $i < $nb; $i++) {
    echo $i;
}

```

→ 0 1 2 3 4

```

for ($i = 0, $nb = 5; $i < $nb; $i++) {
    echo $i;
}

```

→ 0 1 2 3 4

```

for ($i = $a = 0, $b = 1; $i < 7; $i++, $a += $b, $b += $a) {
    echo $a, $b;
}

```

→ 0 1 1 2 3 5 8 13 21 34 55 89 144 233

http://en.wikipedia.org/wiki/Fibonacci_number
francois.piat@univ-fcomte.fr

Instructions
for (2)
PHP 12

Les expressions sont facultatives.

```

$nb = 5;
$i = 0;
for ( ; $i < $nb; $i++) {
    echo $i;
}

```

→ 0 1 2 3 4

```

$nb = 5;
$i = 0;
for ( ; $i < $nb; ) {
    echo $i;
    $i++;
}

```

→ 0 1 2 3 4

```

$i = 0;
for ( ; ; ) {
    echo $i;
    if ($i++ > 4) {
        break;
    }
}

```

→ 0 1 2 3 4

francois.piat@univ-fcomte.fr


Instructions

for (3)


PHP 13

Ne jamais appeler une fonction dans le test d'itération.


```
for ($i = 0; $i < count($tab); $i++) {
    echo $i;
}
```



```
$nb = count($tab);
for ($i = 0; $i < $nb; $i++) {
    echo $i;
}
```



```
for ($i = 0, $nb = count($tab); $i < $nb; $i++) {
    echo $i;
}
```



francois.piat@univ-fcomte.fr

Instructions

for (4)

PHP 14

A utiliser quand l'ordre de traitement n'a pas d'importance.

```
for ($i = 5; $i >= 0; $i--) {
    echo $i;
}
```

→ 5 4 3 2 1 0

```
$t = array('a', 'b', 'c', 'd');
for ($i = count($t) - 1; $i >= 0; $i--) {
    echo $t[$i];
}
```

→ d c b a

```
for ($i = count($t); $i--; ) {
    echo $t[$i];
}
```

→ d c b a

francois.piat@univ-fcomte.fr

Instructions

foreach

PHP 15

Itération sur les éléments d'un tableau

```
foreach (tableau as valeur) {
    instruction(s)
}
```

```
foreach (tableau as cle => valeur) {
    instruction(s)
}
```

```
foreach ($tab as $c => $v) {
    // à l'intérieur de la boucle
    // $c est l'index de l'élément courant
    // $v est la valeur de l'élément courant
}
```

francois.piat@univ-fcomte.fr

Instructions

while

PHP 16

Boucle de traitement.

```
while (expression) {
    instruction(s)
}
```

```
$i = 0;
while ($i < 5) {
    echo $i;
    $i++;
}
```

→ 0 1 2 3 4

```
$i = -1;
while ($i++ < 5) {
    echo $i;
}
```

→ 0 1 2 3 4

```
while ($T = mysqli_fetch_assoc($R)) {
    instructions de traitement
}
```

francois.piat@univ-fcomte.fr

Instructions

continue

PHP 17

continue permet de passer une itération.

```

$notes = array(10, 14, 'abs', 12, 'abs');
$nb = count($notes);
$total = $div = 0;

for ($i = 0; $i < $nb; $i++) {
    if ($notes[$i] === 'abs') {
        continue;
    }
    $total += $notes[i];
    $div++;
}

$moyenne = $total / $div;

```

→ **\$moyenne = 12**

francois.piat@univ-fcomte.fr

Instructions

break

PHP 18

break interrompt une boucle **for** ou **while**

```

$tab = array('Pierre', 'Paul', 'Jacques', 'Jean');
$pos = -1;

for ($i = 0, $nb = count($tab); $i < $nb; $i++) {
    if ($tab[$i] == 'Paul') {
        $pos = $i;
        break;
    }
}

if ($pos > -1) {
    echo "Paul est à la position $pos";
} else {
    echo "Paul n'est pas dans le tableau";
}

```

→ **Paul est la position 1**

francois.piat@univ-fcomte.fr

Instructions

exit

PHP
19

exit arrête le traitement du script.

```
if (! mysqli_connect('url', 'login', 'passe', base)) {  
    exit;  
}
```

```
if (! mysqli_connect('url', 'login', 'passe', base)) {  
    exit('Connexion au serveur de BD impossible');  
}
```

francois.piat@univ-fcomte.fr

Instructions

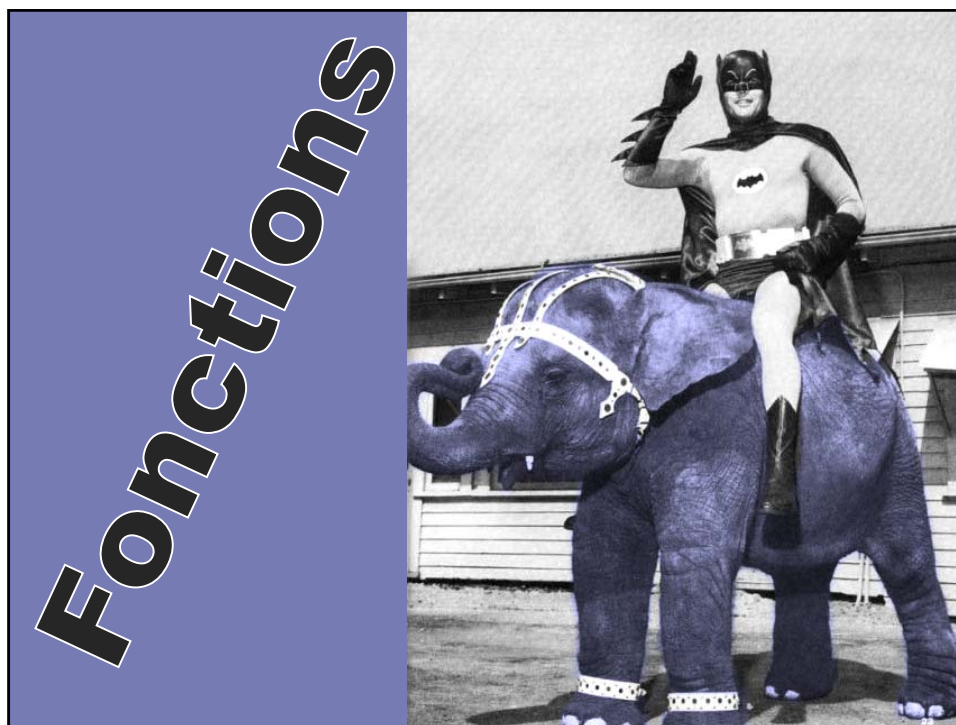
return

PHP
20

return renvoie une valeur définie dans une fonction.

```
function moyenne($notes) {  
    $iMax = count($notes);  
    $total = $nbNotes = 0;  
  
    for ($i = 0; $i < $iMax; $i++) {  
        if ($notes[$i] == 'abs') {  
            continue;  
        }  
        $total += $notes[i];  
        $nbNotes ++;  
    }  
  
    return $total / $nbNotes;  
}
```

francois.piat@univ-fcomte.fr



Fonctions	A la base de PHP	PHP 22
<ul style="list-style-type: none">● PHP repose sur des bibliothèques de fonctions :<ul style="list-style-type: none">● chaînes de caractères● tableaux● accès bases de données● Pas de procédure dans PHP : tout est fonction.● La casse n'a pas d'importance : <code>MaFonction()</code> <code>maFonction()</code> <code>MAFONCTION()</code>● Les fonctions peuvent être définies n'importe où dans le code.		
<small>francois.piat@univ-fcomte.fr</small>		

Fonctions
Définition et invocation
PHP 23

```
function debutHTML () {
    echo '<!DOCTYPE html>',
        '<html>',
        '<head>',
        '<title>Accueil</title>',
        '</head>',
        '<body>';
}
```

```
debutHTML();
```

Résultat de l'invocation : envoyer au navigateur la chaîne de caractères

<html><head><title>Accueil</title></head><body>

francois.piat@univ-fcomte.fr

Fonctions
Paramètres
PHP 24

Une fonction peut avoir 0, 1 ou plusieurs paramètres.

```
function debutHTML($titre, $css) {
    echo '<!DOCTYPE html>',
        '<html>',
        '<head>',
        '<title>', $titre, '</title>',
        '<link href="", $css, "">',
        '</head>',
        '<body>';
}
```

```
debutHTML('Accueil', 'styles.css');
```

```
debutHTML('Produits', 'styles.css');
```

```
debutHTML('Commande', 'caddy.css');
```

francois.piat@univ-fcomte.fr

Fonctions
Valeur par défaut
PHP 25

Les paramètres peuvent avoir une valeur par défaut.

```
function debutHTML($titre = 'Site', $css = 'style.css') {
    echo '<DOCTYPE html>',
        '<html>',
        '<head>',
        '<title>', $titre, '</title>',
        '<link href="", $css, ">',
        '</head>',
        '<body>';
}
```

debutHTML();

debutHTML('Accueil');

debutHTML('Connexion', 'login.css');

francois.piat@univ-fcomte.fr

Fonctions
Paramètres implicites
PHP 26

Les paramètres peuvent ne pas être déclarés.

```
function somme() {
    $nbArgs = func_num_args();

    $total = 0;
    for ($i = 0; $i < $nbArgs; $i++) {
        $total += func_get_arg($i);
    }

    echo $total;
}
```

somme(10, 20, 30);

somme(1, 2, 3, 4, 5, 6, 7, 8);

francois.piat@univ-fcomte.fr

Fonctions
return (1)
PHP 27

return permet de renvoyer une valeur définie dans la fonction.

```
function somme() {
    $nbArgs = func_num_args();

    $total = 0;
    for ($i = 0; $i < $nbArgs; $i++) {
        $total += func_get_arg($i);
    }

    return $total;
}
```

```
$s = somme(10, 20, 30);
echo somme(10, 20, 30);
$m = somme(10, 20, 30) / 3;
uneFonction(somme(10, 20, 30));
```

francois.piat@univ-fcomte.fr

Fonctions
return (2)
PHP 28

- **return** peut renvoyer n'importe quel type de valeur.

```
return 'OUI';
return true;
return 10;

return array($a1, $a2, $a3);

return ($a > 10) ? 'ok' : 'ko';

return somme($a, $b, $c);
```

- **return** peut être utilisé sans renvoyer de valeur.

```
return;
```

francois.piat@univ-fcomte.fr

Fonctions
return (3)
PHP 29

Plusieurs return dans une fonction.

Exemple : calculer une moyenne à partir de notes
 La fonction `moyenne()` accepte en paramètre un tableau
 Le tableau contient des notes
 Si l'étudiant était absent la note est 'abs'
`$notes = array(10, 14, 'abs', 18, 'abs');`
`$m = moyenne($notes);`

```
function moyenne($n) {
    $iMax = count($n);
    for ($i = $total = $nbNotes = 0; $i < $iMax; $i++) {
        if (is_number($notes[$i])) {
            $total += $notes[i];
            $nbNotes ++;
        }
    }
    return $total / $nbNotes;
}
```

francois.piat@univ-fcomte.fr

Fonctions
return (4)
PHP 30

```
function moyenne($n) {
    if (! is_array($n)) {
        return -1; // RETURN : erreur paramètre
    }
    $nb = count($n);
    if ($nb == 0) {
        return -2; // RETURN : tableau vide
    }

    for ($i = $total = $nbNotes = 0; $i < $nb; $i++) {
        if ($notes[$i] === 'abs') {
            continue;
        }
        if (! is_number($notes[$i])) {
            return -3; // RETURN : erreur valeur
        }
        if ($notes[$i] < 0) {
            return -4; // RETURN : erreur valeur
        }
        $total += $notes[i];
        $nbNotes ++;
    }
    return ($nbNotes == 0) ? 'abs' : $total / $nbNotes;
}
```

francois.piat@univ-fcomte.fr

Fonctions	Fonctions variables	PHP 31
<pre>\$Variable = 'nomDeFonction'; \$Variable();</pre> <div style="background-color: #f0f0f0; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>if (function_exists('mysqli_real_escape_string')) { \$Protect = 'mysqli_real_escape_string'; } elseif (function_exists('mysqli_escape_string')) { \$Protect = 'mysqli_escape_string'; } else { exit('Version de PHP trop ancienne'); }</pre> </div> <div style="background-color: #f0f0f0; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>\$nom = \$Protect(\$bd, \$nom); \$prenom = \$Protect(\$bd, \$prenom); \$sql = "INSERT INTO table SET nom = '\$nom', prenom = '\$prenom'";</pre> </div>		
francois.piat@univ-fcomte.fr		

Fonctions	Bibliothèques	PHP 32
<ul style="list-style-type: none"> ● Définir ses propres bibliothèques pour le développement <ul style="list-style-type: none"> ● une ou plusieurs ● gestion de formulaires, ● gestion de code HTML, ● gestion BD ● etc. ● Lier une bibliothèque de fonctions à un script avec <ul style="list-style-type: none"> ● <code>include('bibli.php');</code> ● <code>require('communs/bibli.php');</code> ● <code>include_once('../communs/bibli.php');</code> ● <code>require_once('bibli.php');</code> 		
francois.piat@univ-fcomte.fr		

Fonctions

Decrire une fonction

PHP
33

```
/**
 * Vérification d'un paramètre GET
 *
 * Vérification que des paramètres passés dans un GET (url)
 * sont numériques et d'une valeur minimum.
 * En cas d'erreur, la fonction retourne FALSE
 *
 * @param string $nom      Nom du paramètre à vérifier
 * @param integer $mini     Valeur minimum du paramètre
 * @param boolean $requis   Paramètre obligatoire ou pas
 * @global array $_GET
 *
 * @return integer Valeur du paramètre ou FALSE
 */
function verifGET($nom, $mini = 0, $requis = TRUE) {
    ...
}
```

www.phpdoc.org

francois.piat@univ-fcomte.fr