Cheatography

PHP Syntaxe & Fondamentaux - FR Cheat Sheet

by Zetura (Zetura) via cheatography.com/18426/cs/1752/

Syntaxe de base

Code PHP	<?php //Contenu ?>
Commentaire sur une ligne	// Commentaire
Commentaires sur plusieurs lignes	/* Commentaire */
Fin d'instruction	;
Inclusion de fichier	require_once('nom_fichier.php');

Variables et Constantes

```
$nomVariable = "Chaine de caractere";
$nomVariable = 'Chaine de caractere';
$nomVariable = 5;
$nomVariable = "Une {$autreVariable} info";
echo $nomVariable;
global $varGlobale;
echo $GLOBALS['varGlobale']
define('NOMCONSTANTE', 'valeur');
echo NOMCONSTANTE;
```

Fonctions sur les variables

Vérifier l'existence de la variable	isset($var);
Détruire une variable	unset($var);
Connaître le type	gettype($var);
Vérifier un type	is_[type]($var); Ex : is_string($var);
Cast (changement de type)	$var = (string) $var;
Conversion de valeur	[type]val($var); Ex : intval($var); $floatval($var);
Test si la variable est vide	empty($var);

Tableaux

Création	$tableau = array();
Ajout	$tableau[] = "valeur";
Ajout sur un index	$tableau[4] = "valeur";

Tableaux (cont)

Ajout sur une clé	$tableau["cle"] = "valeur";
Création numérique	$tableau = array('valeur1', 'valeur2');
Création associative	$tableau = array('cle1' => 'valeur1', 'cle2' => 'valeur2');
Ecriture depuis numérique	echo $tableau[4];
Ecriture depuis association	echo $tableau['cle1'];
Tableaux numériques	Les clés sont des chiffres
Tableaux associatifs	Les clés sont des chaînes de caractères
Matrice (tableau multi-dimensions)	$matrice[2][3] = "valeur";

Types

Booléen	boolean	true OU false
Entier	integer	nombre positif ou négatif
Nombre flottant	float	nombre à virgule positif ou négatif
Nombre flottant	double	nombre à virgule positif uniquement
Chaîne de caractère	string	chaîne de caractères

Chaines de caracteres (string)

Délimités entre guillement ou apostrophe	"chaine de caractères" ou 'chaine de caractères'
Entre guillement, les variables sont interprétés	$var = caractères; echo "chaine de $var";
Caractère d'échappement	\
Retour à la ligne	\n
Retour chariot	\r

Chaines de caracteres (string) (cont)

Tabulation	\t

Fonction utilisateur

```
function multiplier($arg1, $arg2)
{
return $arg1 * $arg2;
}
$param1 = 4;
$param2 = 8;
$resultat = multiplier($param1, $param2);
```

Classes and Objects

```
class SomeClass {
    private $property;
    public $anotherProperty;
    protected $yetAnotherProperty = null;
    public function __construct($arg=null)
{
    $this->property = $arg;
}
public function someMethod()
{
    echo “Hi”;
}
public function getProperty()
{
    return $this->property;
}
public function setProperty( $p )
{
    $this->property = $p;
}
}
$myObject = new SomeClass( “123” );
echo $myObject->getProperty(); // 123
$myObject->property; // ERROR:private
```

By **Zetura** (Zetura)
cheatography.com/zetura/
atago.fr

Published 19th February, 2014.
Last updated 2nd June, 2014.

Cheatography

PHP Syntaxe & Fondamentaux - FR Cheat Sheet

by Zetura (Zetura) via cheatography.com/18426/cs/1752/

Operateurs

Affectation	=	$var = 5;
Affectation par référence	&=	$nouvelVar &= $var;
Addition	+	$var = $var + 5;
Soustraction	-	$var = $var - 5;
Multiplication	*	$var = $var * 5;
DIvision	/	$var = $var / 5;
Modulo	%	$var = $var % 5;
Incrémentation	++	$var = $var++;
Décrémentation	--	$var = $var--;
Opérateurs combinés	[opérateur]=	$var += 5; $var *= 5;
Concaténation	.	echo $var." chaine";
Concaténation et assignation	.=	$var .= " chaine";

Operateurs de comparaison

Egalité	==	if($var == 5)
Inférieur à	<	if($var < 5)
Inférieur ou égal à	<=	if($var <= 5)
Supérieur à	>	if($var > 5)
Supérieur ou égal à	>=	if($var >= 5)
Différent de	!=	if($var != 5)
Strictement égal (valeur et type)	===	if($var === 5)
Différent en valeur ou en type	!==	if($var !== 5)

UtlisÃ©s pour les structures conditionnelles

Operateurs logiques

Inversion	!	Retourne true si false, et inversement
ET	&&	Retourne true si 2 conditions à true
Un seul	^	Retourne true si une seule des conditions à true
OU	||	Retourne true si une condition à true
ET non prioritaire	AND	Similaire à && mais moins prioritaire
Un seul non prioritaire	XOR	Similaire à ^ mais moins prioritaire
OU non prioritaire	OR	Similaire à || mais moins prioritaire

UtilisÃ©s pour les structures conditionnelles

Fonctions PHP utiles

Récupérer une partie d'une chaîne	substr($string, start, length);
Transformer une chaine en tableau	explode(',', $string);
Concaténer un tableau en chaine	implode(',', $tableau);
Retirer les espaces au début et à la fin d'une chaîne	trim($string);
Remplacer à par b dans une chaine	str_replace('a', 'b', $string);
Vérifier une expression régulière	preg_match('regex', $string);
Remplacer une expression régulière par b	preg_replace('regex', 'b', $string);
Arrêter le script PHP	exit();
Envoyer un mail	mail($mailDest, $sujet, $message, 'From: '.$mailEnvoi);

Expression reguliere

^	Début de chaîne
\d	Chiffre entre 0 et 9
\w	Caractère alphanumérique [0-9A-Za-z]
\s	Espace
.	N'importe quelle lettre, chiffre ou espace
$	Fin de chaîne
()	Groupe
[]	Classe de caractères
{x} {x,} {x,y}	Quantité = x | Supérieur ou égal à x | Entre x et y
*	Quantité de 0 ou plus
?	Quantité de 0 ou 1
+	Quantité de 1 ou plus
|	OU
\	Caractère d'échappement

Exemple pour une syntaxe de mail : ^[\w.-\+]+@[\w.-]+\.[a-zA-Z]{2,6}$

Structure conditionnelle : IF

```
if (condition) {
// Instructions
}
elseif (condition) {
// Instructions
}
else {
// Instructions
}
if( $something == true ) {
// Si $something vaut true
doSomething();
} elseif ( $something == false ) {
// Si $something vaut false
doSomethingElse();
} else {
// sinon, exécuter doNothing();
doNothing();
}
if(condition):
```

By **Zetura** (Zetura)
cheatography.com/zetura/
atago.fr

Published 19th February, 2014.
Last updated 2nd June, 2014.

Structure conditionnelle : IF (cont)

```
// Instructions
endif;
(condition)? instructions si true : instructions si false;
```

Structure conditionnelle : SWITCH

```
switch ($var) {
case 1:
// Instructions
break;
case "test":
// Instructions
break;
default:
// Instructions
break;
}
```

Peut Ãªtre utilisÃ© avec des chiffres ou chaÃ®nes de caractÃ¨res

Boucle WHILE

Tant que la condition est vrai, l'instruction est exécutée

```
while(condition){
// Instructions
}
$i = 1;
while($i < 10){
echo $i;
$i++;
}
```

Exécution au moins une première fois

```
$i = 1;
do{
echo $i;
$i++;
}
while($i < 10);
```

Attention aux boucles infinies

Boucle FOR

Exécute la première expression lors de l'initialisation, puis tant que la condition est valide, exécute le contenu de la boucle et fini en exécutant la dernière expression

```
for(expression1; condition; expression2){
// Instructions
}
for($i = 1; $i < 10; $i++){
// Instructions
}
```

Boucle FOREACH

A chaque itération dans la boucle assigne la valeur de l'élément courant à la variable et le pointeur interne du tableau est avancé d'un élément.

```
foreach($tableau as $element){
// Instructions
}
foreach($tableau as $key => $value){
// Instructions
}
$tableau = array(
'prenom' => 'Obi-wan',
'nom' => 'Kenobi',
'metier' => 'Jedi'
);
foreach($tableau as $contenu){
echo "Valeur : $contenu<br/>";
}
foreach($tableau as $cle => $valeur){
echo "Clé : $cle -> Valeur : $valeur<br/>";
}
```

Attention, la boucle fonctionne sur une copie du tableau spÃ©cifiÃ©, pas sur le tableau lui-mÃªme

CONTINUE

```
for ($i = 0; $i < 5; ++$i) {
if ($i == 2)
continue;
print "$i , ";
}
```

produces the following output:
0 , 1 , 3 , 4

By **Zetura** (Zetura)
cheatography.com/zetura/
atago.fr

Published 19th February, 2014.
Last updated 2nd June, 2014.