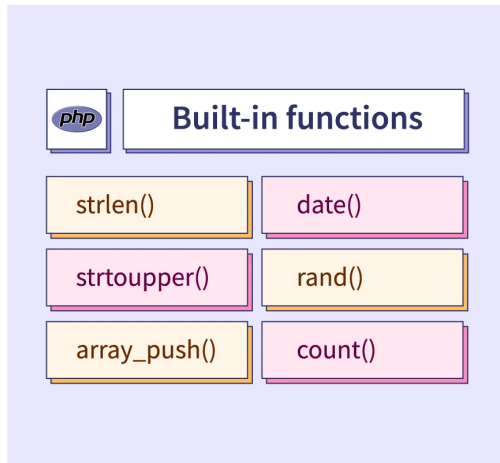


# Les Fonctions Internes en PHP



En PHP, les fonctions internes (ou intégrées) sont des fonctions fournies par le langage lui-même et qui permettent de réaliser une multitude de tâches courantes sans avoir besoin de définir vos propres fonctions. Voici un aperçu des principales catégories de fonctions internes en PHP ainsi que quelques exemples pour chacune :

## 1. Fonctions de chaîne (String Functions)

Function Name	Parameters	Description
strlen	One string	Returns number of characters in the string
strcmp	Two strings	Returns zero if both strings are equal, a -ve number if the first string occurs before second string or a +ve number if the first string occurs after the second string
strpos	Two strings	Returns position of second string in the first string or false if not found
substr	One string and one integer	Returns the substring from the specified string from the position specified as an integer. If a third integer value is specified, it represents the length of the substring to be retrieved
chop	One string	Returns the string with all white space characters removed from the end
trim	One string	Returns the string with all white space characters removed on both sides
ltrim	One string	Returns the string with all white space characters removed from the beginning
strtolower	One string	Returns the string with all the characters converted to lowercase
strtoupper	One string	Returns the string with all the characters converted to uppercase
strrev	One string	Returns the reverse of the given string
str_replace	Three strings	Returns the string in which a old substring is replaced by the new substring
str_word_count	One string	Returns the word count in the given string

Ces fonctions permettent de manipuler les chaînes de caractères. -

`strlen($string)` : Retourne la longueur d'une chaîne. - `strpos($haystack, $needle)` : Cherche la position de la première occurrence d'une sous-chaîne.  
- `str_replace($search, $replace, $subject)` : Remplace toutes les occurrences d'une sous-chaîne par une autre.

```
$length = strlen("Bonjour");
echo $length; // Affiche 7
```

```
$position = strpos("Hello World", "World");
echo $position; // Affiche 6
```

```
$newString = str_replace("World", "PHP",
"Hello World");
echo $newString; // Affiche Hello PHP
```

## 2. Fonctions de tableau (Array Functions)

Function Name	Parameters	Description
<code>count</code>	One array	Returns the number of elements in the array
<code>unset</code>	One array	Deletes the element from memory
<code>array_keys</code>	One array	Returns the keys (indexes) as an array
<code>array_values</code>	One array	Returns the values as an array
<code>array_key_exists</code>	Key, array	Returns <i>true</i> if the key is found, <i>false</i> otherwise
<code>is_array</code>	One array	Returns <i>true</i> if the argument is an array, <i>false</i> otherwise
<code>in_array</code>	Exp, array	Returns <i>true</i> if <i>exp</i> is in the array, <i>false</i> otherwise
<code>explode</code>	Delim, string	Returns an array of substrings based on the <i>delim</i>
<code>implode</code>	Delim, array	Returns a string joining the array elements with <i>delim</i>
<code>sort</code>	One array	Sorts the values in the array and renames the keys to integer values 0, 1, 2, ...
<code>asort</code>	One array	Sorts the values in the array preserving the keys
<code>ksort</code>	One array	Sorts the keys in the array preserving the values
<code>rsort</code>	One array	Reverse of sort (descending order)
<code>rasort</code>	One array	Reverse of asort (descending order)
<code>rksort</code>	One array	Reverse of ksort (descending order)

---

Ces fonctions permettent de manipuler les tableaux. - `array_merge($array1, $array2)` : Fusionne deux tableaux. - `array_push($array, $value)` : Ajoute un ou plusieurs éléments à la fin d'un tableau. - `in_array($needle, $haystack)` : Vérifie si une valeur existe dans un tableau.

```
$array1 = [1, 2, 3];
$array2 = [4, 5, 6];
$result = array_merge($array1, $array2);
print_r($result); // Affiche [1, 2, 3, 4, 5, 6]

array_push($array1, 4, 5);
```

```
print_r($array1); // Affiche [1, 2, 3, 4, 5]

$exists = in_array(3, $array1);
echo $exists; // Affiche 1 (true)
```

### 3. Fonctions de gestion des fichiers (File Handling Functions)

Function	Operation
fopen()	Creates a new file / opens an existing file
fclose()	Closes a file which has been opened for use
getc()	Reads a character from the file
putc()	Writes a character to the file
fprintf()	Write data values to a file
fscanf()	Reads a set of data values from a file
getw()	Reads an integer from the file
putw()	Writes an integer to a file
fseek()	Sets the position to the desired point in the file
ftell()	Gives the current position in the file
rewind()	Sets the position to the beginning of the file

---

Ces fonctions permettent de manipuler les fichiers. - `fopen($filename, $mode)` : Ouvre un fichier. - `fwrite($handle, $string)` : Écrit dans un fichier. - `fclose($handle)` : Ferme un fichier.

```
$file = fopen("test.txt", "w");
fwrite($file, "Hello, World!");
fclose($file);
```

#### 4. Fonctions de date et heure (Date and Time Functions)

```
1 <?php
2 echo date('m/d/y') # 08/04/22
3 . date('m/d/Y') # 08/04/2022
4 . date('d m Y') # 08/04/2022
5 . date('d M y') # 04 Aug 22
6 . date('d M Y') # 04 Aug 2022
7 . date('H:i:s a') # 07:48:01 am
8 . date('H:i:s A') # 07:48:01 AM
9 . date('d M y H:i:s A'); # 04 Aug 22 07:48:01 AM
10
11 echo date ('D\ay: d, \M\o\n\t\h: M, \Y\ea\r: Y');
12 # Prints: Day: 04, Month: Aug, Year: 2022
13
14 $ts = 1659514532;
15 echo date('d M Y - H:i:s', $ts); # Prints: 03 Aug 2022 - 10:15:32
16 echo date('m-d-Y', $ts); # Prints: 08-03-2022
17
18 echo date ( strtotime('December 13, 99') ); #Prints: 12-13-1999
19
20 echo idate('Y'); #Prints: 2022
21 echo idate('m'); #Prints: 8
22
23 echo gmdate('d M Y H:i:s', 0); # Prints: 01 Jan 1970 00:00:00
24
```

Formatting  
Date and  
Time

date()  
idate()  
gmdate()

Ces fonctions permettent de manipuler les dates et les heures. - `time()` : Retourne l'heure actuelle en tant que timestamp Unix. - `date($format, $timestamp)` : Formate une date/heure locale. - `mktime($hour, $minute, $second, $month, $day, $year)` : Retourne le timestamp Unix pour une date donnée.

```
$timestamp = time();
echo $timestamp; // Affiche un timestamp Unix
```

```
$date = date("Y-m-d H:i:s", $timestamp);
echo $date; // Affiche la date et l'heure actuelles
```

```
$customDate = mktime(0, 0, 0, 12, 31, 2024);
echo date("Y-m-d", $customDate); // Affiche
//2024-12-31
```

## 5. Fonctions mathématiques (Math Functions)

PHP Math functions	
➤ <b>abs()</b> : Returns absolute value.	➤ <b>min()</b> : Returns lowest value from given val.
➤ <b>ceil()</b> : Returns int value $\geq$ X.	➤ <b>max()</b> : Returns highest value from given val.
➤ <b>floor()</b> : Returns int value $\leq$ X.	➤ <b>pow()</b> : Returns the val. of X power Y.
➤ <b>round()</b> : Rounding no. nearest to X.	➤ <b>sqrt()</b> : Returns the square root of X.
➤ <b>fmod()</b> : Return the remainder of division.	➤ <b>rand()</b> : Generate a random no. between specified range.(Min & Max val.)

Ces fonctions permettent de réaliser des opérations mathématiques. - `abs($number)` : Retourne la valeur absolue d'un nombre. - `round($number, $precision)` : Arrondit un nombre à la précision spécifiée. - `rand($min, $max)` : Génère un nombre aléatoire entre les valeurs spécifiées.

```
echo abs(-10); // Affiche 10
```

```
echo round(3.14159, 2); // Affiche 3.14
```

```
echo rand(1, 100); // Affiche un nombre aléatoire  
//entre 1 et 100
```

## 6. Fonctions de manipulation des variables (Variable Handling Functions)

PHP VARIABLE FUNCTIONS	
<code>string gettype( \$var );</code>	gets the data type of the variable
<code>bool isset( \$var );</code>	checks if the variable has been initialized
<code>void unset( \$var );</code>	destroys variable
<code>bool is_float( \$var );</code>	checks if variable type is float
<code>bool is_int( \$var );</code>	checks if variable type is integer
<code>bool is_null( \$var );</code>	checks if variable type is null
<code>bool empty( \$var );</code>	check if variable is null or doesn't exist
<code>bool is_numeric( \$var );</code>	checks if variable value is a number
<code>string var_dump( \$var );</code>	provides information about a variable
<code>int intval( \$var );</code>	converts variable value to integer type
<code>string strval( \$var );</code>	converts variable value to string type
<code>bool is_array ( \$var );</code>	checks if variable data type is array
<code>bool settype ( \$var, string \$type );</code>	set the type of a variable

Ces fonctions permettent de manipuler et de vérifier les variables. - `isset($var)` : Vérifie si une variable est définie. - `empty($var)` : Vérifie si une variable est vide. - `is_array($var)` : Vérifie si une variable est un tableau.

```
$var = "PHP";  
echo isset($var); // Affiche 1 (true)
```

```
$emptyVar = "";  
echo empty($emptyVar); // Affiche 1 (true)
```

```
$array = [1, 2, 3];  
echo is_array($array); // Affiche 1 (true)
```

### Conclusion

PHP propose un grand nombre de fonctions internes dans des domaines variés, ce qui facilite grandement le développement en offrant des solutions prêtes à l'emploi pour de nombreuses tâches courantes. Pour une liste complète des fonctions, la documentation officielle de PHP est une ressource précieuse.