Cheatography

PHP Cheat Sheet

by Syshella via cheatography.com/128892/cs/34652/

Strings and variables

Print as html	echo "This is my string";	This is my string
String	"Hello World!";	Hello World!
Escape sequence	"Hello \"World\"!";	Hello "-World"!
Concatenation (Literal)	echo "one" . "two";	onetwo
Variable creation and assignment	$var = "Hello";	
Variable contat-enation	echo $var . " you";	Hello you
Variable parsing	echo "$var you";	Hello you
Expanding string	echo "${v-ar}oo! you!;	Hellooo! you!
Concatenating assigmente	$var .= " world";	Hello world
Assign by reference	$var1 = $var2;	

Types and operators

String	$string = "Hello";
Int	$int = 4;
Float	$float = 4.2;
Exponentiation	4**2 --> (4^2) = 64
Modulo	7 % 3 --> 1
Mathematical assignment operator	$int += 3;
Increment and decrement	$int ++;

Types and operators (cont)

Global variable	global %var; //Inside a function to access a value declared outside the function

- Result of adding up two floats 9.9 and 1.0 will return an integer, since the result is 10 and evaluates to a whole number. Applies to every calculation. The reverse is also true.
- Operations order: () >> * >> and / >> + and -

Functions

```
function greeting($value = "Sam") // Default value = Sam
{
        return " Hello $value !";
}
$retur n_value = greeti ng( " -Tom ");
echo $retur n_v alue; //Prints: Hello Tom!
echo greeti ng(); //Prints: Hello Sam!
function addXPe rma nently (& -$param) // Passed by reference
{
    $param = $param . " X";
   echo $param;
};
$word = " Hel lo";
addXPe rma nen tly ($w ord); // Prints: HelloX
echo $word; // Prints: HelloX
```

Built-in Functions

gettype(4); // Prints: integer	Get the type of the parameter given as a string
var_dump(4); // Prints: int(10-00000)	Prints details about the argument
strive("Hello"); // Prints: olleH	Prints the string in reversed order

Built-in Functions (cont)

strtolower("Hell-o"); // Prints: hello	Lower case the string
str_repeat("Hell-o", 2); // Prints: HelloHello	Repeats the string the number of times specified
substr_count-($string, $subst-ring);	Number of instances of a substring within a string
abs(-2); // Returns 2	Returns the absolute value
round(3.8); // Returns 4	Rounds a number
rand();	Returns a random number between 0 and the largest allowed
rand(3, 10);	Returns a number between 3 and 10 inclusive
getrandmax()	Returns the largest random number value possible

Data Structures

Array	$my_array = array(0, 1, "A");
Array	$my_array = [0, 1, "A"];
Print array	print_r($my_array);
Returns array separated by value	implode(", ", $my_array);
Access array index	$y_array[1];
Adds the element at the end of the array	$my_array = "new element";
Replace the element	$my_array[0] = "new first elemen-t";
Pops the last element and returns it	array_pop($my_a-rray);

Cheatography

PHP Cheat Sheet

by Syshella via cheatography.com/128892/cs/34652/

Data Structures (cont)

Push the elements to the end of the array and return its size	array_push("A", "B");

count($array); // Returns the number of elements in the array

- Different types are allowed in an array

Map

Associative array

```
$my_array = ["pa nda " => "very cute", " liz ard " => " cut e", " coc kro ach " => "not very cute"];
```

Array function

```
$about_me = array(
        " ful lna me" => " Aisle Nevert ell ",
        " soc ial " => 123456789
);
echo implod e($ my_ array); // Prints only the values
print_ r($ my_ array); // Prints keys and values
$my_array = ["ne w"] = "new item"; // Adds new element
$favorites = ["fa vor ite _fo - od"= >"pi zza ", " fav ori te_ - pla ce"= >"my dreams ", " FAV - ORI TE_ CAS E"=> " CAP S", " - fav ori te_ per son " =>"m yse - lf"];

echo $favor ite s["f avo rit e" . " _" . " foo d"];
// Prints: pizza

$key = " fav ori te_ pla ce";
echo $favor ite s[$ key];
// Prints: my dreams

echo $favor ite s[s trt oup - per ("fa vor ite _ca se")];
// Prints: CAPS
unset( " my_ arr ay[ " new "]);
// Delete the " new " element if exists
```

key => value

Libraries

<?php include 'footer.html';?>	Includes the HTML file where specified
<?php require 'somefile.php'; ?>	Requires the specified file

PHP & HTML

Data access in Forms

$_GET	Non sensible information
$_POST	Used to send sensitive information
$_GET['user']	Retrieves the value sent through the user get form
$_isset($-_POST['se-nd'])	Check if send has a value

When a form is submitted values are received contained in an associative array.

Works as a dictionary. Arrays are maps where the key is the index.
Key can be number or characters.

```
<?php
$lucky _number = 5 * 2 - 1;

echo " <h1 >Your lucky number is
${luck y_n umb er} </h 1>";
?>
<?php
function makeHe ade rGr eeting
($name){
    return " <h1 >Hello,
${name }!< /h1 >";
}

echo makeHe ade rGr eet ing -
("Wo rld ");
?>
$about_me = [
    " nam e" => " Aisle Nevert -
ell ",
    " bir th_ yea r" => 1902,
    " fav ori te_ foo d" => " -
piz za"
];
function calcul ateAge ($pers -
on_ arr){
    $cu rre nt_year =
date("Y ");
   $age = $curre nt_year -
$perso n_a rr[ " bir th_ yea -
r"];
    return $age;
}
?>
<h1 >We lco me! </h 1>
<h2 >About me:</h 2>
<?php
   echo " <h3 >Hello! I'm
{$abou t_m e["n ame " ]}! </h -
3>";
   echo " <p> I'm " . calcul -
ate Age ($a bou t_me). " years
old! That's pretty cool,
right? </p >";
   echo " <di v>What more is
there to say? I love {$abou -
t_m e["f avo rit e_f ood "]},
and that's pretty much it!</d -
iv> ";
?>
```