

Strings and variables

Print as html	<code>echo "This is my string";</code>	This is my string
---------------	--	-------------------

String	<code>"Hello World!";</code>	Hello World!
--------	------------------------------	--------------

Escape sequence	<code>"Hello \"World!\"";</code>	Hello "World!"
-----------------	----------------------------------	----------------

Concatenation (Literal)	<code>echo "one" . "two";</code>	onetwo
-------------------------	----------------------------------	--------

Variable creation and assignment	<code>\$var = "Hello";</code>	
----------------------------------	-------------------------------	--

Variable concatenation	<code>echo \$var . "you";</code>	Hello you
------------------------	----------------------------------	-----------

Variable parsing	<code>echo "\$var you";</code>	Hello you
------------------	--------------------------------	-----------

Expanding string	<code>echo "\${var}oo! you!";</code>	Hellooo! you!
------------------	--------------------------------------	---------------

Concatenating assignment	<code>\$var .= "world";</code>	Hello world
--------------------------	--------------------------------	-------------

Assign by reference	<code>\$var1 = \$var2;</code>	
---------------------	-------------------------------	--

Types and operators

String	<code>\$string = "Hello";</code>	
--------	----------------------------------	--

Int	<code>\$int = 4;</code>	
-----	-------------------------	--

Float	<code>\$float = 4.2;</code>	
-------	-----------------------------	--

Exponentiation	<code>4**2 --> (4^2) = 64</code>	
----------------	-------------------------------------	--

Modulo	<code>7 % 3 --> 1</code>	
--------	-----------------------------	--

Mathematical assignment operator	<code>\$int += 3;</code>	
----------------------------------	--------------------------	--

Increment and decrement	<code>\$int ++;</code>	
-------------------------	------------------------	--

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 -`

Functions

```
function greeting($value = "Sam") // Default value = Sam
{
    return " Hello $value !";
}
$return_value = greeting( " - Tom ");
echo $return_value; //Prints: Hello Tom!
echo greeting(); //Prints: Hello Sam!
function addXPercently (& $param) // Passed by reference
{
    $param = $param . " X";
    echo $param;
};
$word = " Hello";
addXPercently ($word); // Prints: HelloX
echo $word; // Prints: HelloX
```

Built-in Functions

<code>gettype(4); // Prints: integer</code>	Get the type of the parameter given as a string
---	---

<code>var_dump(4); // Prints: int(100000)</code>	Prints details about the argument
--	-----------------------------------

<code>strrev("Hello"); // Prints: olleH</code>	Prints the string in reversed order
--	-------------------------------------

Built-in Functions (cont)

<code>strtolower("Hello"); // Prints: hello</code>	Lower case the string
--	-----------------------

<code>str_repeat("Hello", 2); // Prints: HelloHello</code>	Repeats the string the number of times specified
--	--

<code>substr_count(\$string, \$substring);</code>	Number of instances of a substring within a string
---	--

<code>abs(-2); // Returns 2</code>	Returns the absolute value
------------------------------------	----------------------------

<code>round(3.8); // Returns 4</code>	Rounds a number
---------------------------------------	-----------------

<code>rand();</code>	Returns a random number between 0 and the largest allowed
----------------------	---

<code>rand(3, 10);</code>	Returns a number between 3 and 10 inclusive
---------------------------	---

<code>getrandmax();</code>	Returns the largest random number value possible
----------------------------	--

Data Structures

Array	<code>\$my_array = array(0, 1, "A");</code>
-------	---

Array	<code>\$my_array = [0, 1, "A"];</code>
-------	--

Print array	<code>print_r(\$my_array);</code>
-------------	-----------------------------------

Returns array separated by value	<code>implode(" ", \$my_array);</code>
----------------------------------	--

Access array index	<code>\$my_array[1];</code>
--------------------	-----------------------------

Adds the element at the end of the array	<code>\$my_array = "new element";</code>
--	--

Replace the element	<code>\$my_array[0] = "new first element";</code>
---------------------	---

Pops the last element and returns it	<code>array_pop(\$my_array);</code>
--------------------------------------	-------------------------------------



By **Syshella**
cheatography.com/syshella/

Not published yet.
Last updated 19th October, 2022.
Page 1 of 3.

Sponsored by **ApolloPad.com**
Everyone has a novel in them. Finish
Yours!
<https://apollopad.com>

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 = ["panda" => "very cute", "lizard" => "cute", "crocodile" => "not very cute"];
```

Array function

```
$about_me = array(
    "full_name" => "Aisle Nevertell",
    "social" => 123456789
);
```

```
echo implode($my_array); // Prints only the values
print_r($my_array); // Prints keys and values
```

```
$my_array["new"] = "new item"; // Adds new element
$favorites = ["favorite_food" => "pizza", "favorite_place" => "my dreams", "favorite_case" => "CAPS", "favorite_person" => "myself"];
```

```
echo $favorites["favorite_food"]; // Prints: pizza
```

```
$key = "favorite_place";
echo $favorites[$key]; // Prints: my dreams
```

```
echo $favorites["favorite_case"]; // Prints: CAPS
```

```
unset($my_array["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['send']) 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
${lucky_number} </h1>";
?>

<?php
function makeHello greeting
($name){
    return " <h1 >Hello,
$name! </h1 >";
}

echo makeHello greeting (
"World ");
?>

$about_me = [
    "name" => "Aisle Nevertell",
    "birth_year" => 1902,
    "favorite_food" => "piza"
];
function calculateAge ($person_arr){
    $current_year =
date("Y ");
    $age = $current_year -
$person_arr["birth_year"];
    return $age;
}
?>
<h1 >Welcome! </h1 >
<h2 >About me:</h2 >
<?php
    echo " <h3 >Hello! I'm
${about_me["name"]}! </h3 >";
    echo " <p > I'm " . calculateAge ($about_me) . " years
old! That's pretty cool,
right? </p >";
    echo " <div >What more is
there to say? I love {about_me["favorite_food"]},
and that's pretty much it!</div > ";
?>
```

