Function Name	Parameter	Example
boolval	**var**	<?php echo '0:'.(boolval(0) ? 'true' : 'false')."\n"; ?> Output 0: false
empty	Var	$expected_array_got_string = 'somestring'; var_dump(empty ($expected_array_got_string[0])); Output: bool(false)
debug_zval_dump	**variable**	$var1 = 'Hello World'; $var2 = ''; $var2 =& $var1; debug_zval_dump($var1); Output: string(11) "Hello World" refcount(1)
var_dump	**Expression(**The variable you want to dump**)**	$b = 3.1; $c = true; var_dump($b, $c); Output: float(3.1) bool(true)
get_resource_type	**handle**	$fp = fopen("foo", "w"); echo get_resource_type($fp) . "\n"; Output: stream

gettype	The variable being type checked	$data = array(1, 1., NULL, new stdClass, 'foo');
		
		foreach ($data as $value) {
		  echo gettype($value), "\n";
		}
		
		Output:
		integer double NULL object string

intval	**Var (**The scalar value being converted to an integer) **base** The base for the conversion	`echo intval(4.2);` `Output: 4`

floatval	May be any scalar type. floatval() should not be used on objects, as doing so will emit an **E_NOTICE** level error and return 1.	`$var = '122.34343The';`
		`$float_value_of_var = floatval($var);`
		`echo $float_value_of_var;`
		Output
		122.34343

is_bool	**Var**
	The variable being evaluated
		$a = false;
		$b = 0;
		if (is_bool($a) === true) {
		echo "Yes, this is a boolean";
		}
		
		if (is_bool($b) === false) {
		echo "No, this is not a boolean";
		}
		
		Output:
		Yes, this is a booleanNo, this is not a boolean

is_float	The variable being evaluated	`var_dump(is_float(27.25));`
		
		`Output:`
		bool(true)

is_int	The variable being evaluated	function isInteger($input){
		return(ctype_digit(strval($input)));
		}
		
		var_dump(is_int(23));
		var_dump(is_int("23"));
		
		Output:
		bool(true) bool(false)

is_iterable	**Var**
	The value to check
		`var_dump(is_iterable([1, 2, 3]));`
		`var_dump(is_iterable(1));`
		
		`Output:`
		`bool(true)  bool(false)`

isset	**Var** The variable to be checked	$a = "test"; $b = "anothertest";

		var_dump(isset($a)); // TRUE
		var_dump(isset($a, $b)); // TRUE
		
		Output:
		
		bool(true) bool(true)

is_string	The variable being evaluated.	$values = array(false,0);
		foreach ($values as $value) {
		echo "is_string(";
		var_export($value);
		echo ") = ";
		echo var_dump(is_string($value));
		}
		
		Output
		is_string(false) = bool(false) is_string(0) = bool(false

settype	**Var**
	The variable being converted
		$var = "true";
		settype($var, 'bool');
		var_dump($var);
		Output:
		bool(true)

strrev	The string to be reversed.	echo strrev("Hello world!"); Output: "!dlrow olleH"
strlen	**String** The string being measured for length.	$str = ' ab cd '; echo strlen($str); Output: 7
stristr	**Haystack** The string to search in	$email = 'USER@EXAMPLE.com'; echo stristr($email, 'e'); Output: ER@EXAMPLE.com
strip_tags	**Str** The input string	$text = ' Test paragraph. <!-- Comment -> > Other text'; echo strip_tags($text); echo "\n"; Output: Test paragraph. Other text
substr_replace	**string** The input string.	$var = 'ABCDEFGH:/MNRPQR/'; echo "Original: $var<hr />\n"; echo substr_replace($var, 'bob', 0) . " \n"; echo substr_replace($var, 'bob', 0, strlen($var)) . " \n"; Output: bob bob

substr_count	**Haystack**
	Needle
	Offset
	length
		$text = 'This is a test';
		echo strlen($text); // 14
		
		echo substr_count($text, 'is');
		
		Output:
		2

strtolower	**string**	$str = "Mary Had A Little Lamb and She LOVED $str = strtolower($str); echo $str; Output: mary had a little lamb and she loved it so
implode	**Glue** Defaults to an empty string **Pieces** The array of strings to implode.	$array = array('lastname', 'email', 'phone'); $comma_separated = implode(",", $array); echo $comma_separated; var_dump(implode('hel Output: lastname,email,phonestring(0) ""
explode	**Delimiter** The boundary string. **String** The input string.	$pizza = "piece1 piece2 piece3 piece4 piece5 $pieces = explode(" ", $pizza); echo $pieces[0]; echo $pieces[1]; output: piece1piece2

md5	**Str**
	The string
		$str = 'apple';
		if (md5($str) === '1f3870be274f6c49b3e31a0c67
		echo "Would you like a green or red apple
		}
		Output:
		Would you like a green or red apple?

hash	**Algo** Name of selected hashing algorithm (e.g. "md5", "sha256", "haval160,4", etc..) **Data** Message to be hashed.	echo hash('ripemd160', 'The quick brown fox jumped over the lazy dog Output: ec457d0a974c48d5685a7efa03d137dc8bbde7e3
crypt	**Str** The string to be hashed	$password = 'mypassword'; $hash = crypt($password)

sha1	**Str**
	The input string
		`$str = 'apple';`
		
		`if (sha1($str) === 'd0be2dc421be4fcd0172e5afc`
		`    echo "Would you like a green or red apple`
		`}`

sscanf	**str**
	
	The input string being parsed
		`list($serial) = sscanf("SN/2350001", "SN/%d")`
		`$mandate = "January 01 2000";`
		`list($month, $day, $year) = sscanf($mandate,`
		`echo "Item $serial was manufactured on: $year`
		`" . substr($month, 0, 3) . "-$day\n";`
		
		`output:`
		Item 2350001 was manufactured on: 2000-Jan-1

trim	**Str**
	The string that will be trimmed.
		`function trim_value(&$value)`
		`{`
		`    $value = trim($value);`
		`}`
		
		`$fruit = array('apple','banana ', ' cranberry`
		`var_dump($fruit);`
		
		`array_walk($fruit, 'trim_value');`
		`var_dump($fruit);`