

### Arrays

#### Normal array

```
$myGadgets = array( 'iMac',
'iPad', 'iPhone');
```

#### Shorthand array

```
$myGadgets = ['iMac', 'iPad',
'iPhone'];
```

#### Associative array

```
$myGadgets = array( 'computer' => 'iMac', 'tablet' => 'iPad', 'phone' => 'iPhone');
echo $myGadgets['phone'];
(will output iPhone)
```

#### Append to an array

```
$myGadgets[] = 'sound dock';
```

#### Examine contents and structure of array

```
print_r($myGadgets);
```

#### Output a value from an array

```
echo $myGadgets[1];
(Will output 'iPad')
```

#### Convert entire array to string

```
echo implode(' ', $myGadgets);
(First argument is how you would like to separate each element)
```

Arrays are a good way to store several related items of data such as personal details (name, address, DOB)

### Variables

#### Variables can store strings and numbers

```
$txt = "Hello world! ";
$x = 5;
$y = 10.5;
```

#### Variables can be outputted in the following way

```
$txt = "Will";
echo "My name is " . $txt . " !";
```

#### Use a global variable inside a function

```
$x = 5;
$y = 10;
function myTest() {
    global $x, $y;
    $y = $x + $y;
}
```

### Loops

### Loops (cont)

*(The first parameter is the initial value of the variable, the second value controls how many times the loop will execute)*

#### foreach loop

```
$myGadgets = array( 'iMac',
'iPad', 'iPhone');
foreach ($myGadgets AS $myGadget) {
    echo "<li>" . $myGadget . "</li>";
}
```

*(Outputs each element of the array as a list item)*

#### while loop

```
$x = 1;
while(x <= 5){
    echo "Hello World";
    x++;
}
(Loop continues until condition is no longer true)
*do while loop
$x = 1;
do {
    echo "Hello World";
    $x++;
} while ($x <= 5);
(Identical to a while loop except the code is executed once regardless of if the condition is met)
```

**while** - loops through a block of code as long as the specified condition is true

**do while** - loops through a block of code once, and then repeats the loop as long as the specified condition is true

**for** - loops through a block of code a specified number of times

**foreach** - loops through a block of code for each element in an array

### String functions

### Miscellaneous

#### Escaping characters

When using double quotes for example "hello world" you can escape characters with a backslash \

```
break; (stops loop)
exit; (stops php script)
```

### Date and Time

```
h : 12 hr format
H : 24 hr format
i : Minutes
s : Seconds
u : Microseconds
a : Lowercase am or pm
l : Full text for the day
F : Full text for the month
j : Day of the month
S : Suffix for the day
Y : 4 digit year
echo date('h:i:s');
```

### Post

```
$firstName = $_POST['fname'];
```

### for loop

```
for ($x = 5; $x <= 100; $x++) {  
    echo "The number is " . $x;  
}
```

```
explode(string_separator, $str,  
limit);  
implode(string_separator,  
array);  
str_replace(find, replace,  
string, count);  
str_word_count(string,  
return, char);  
strlen($str);  
strtolower($str);  
strtoupper($str);  
trim($str, chars);  
wordwrap($str, width, break,  
cut);  
substr($str, start, length)
```

Not all parameters required.



By [williamdavies](#)

[cheatography.com/williamdavies/](http://cheatography.com/williamdavies/)

Not published yet.

Last updated 13th May, 2016.

Page 1 of 2.

---

Sponsored by [CrosswordCheats.com](http://CrosswordCheats.com)

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>