`sort()` - Sorts an array in ascending order.

```
$numbers = array(3, 1, 4, 1, 5);
sort($numbers);
```

`rsort()` - Sorts an array in descending order.

```
$numbers = array(3, 1, 4, 1, 5);
rsort($numbers);
```

`asort()` - Sorts an array and maintains index association.

```
$fruit = array("key1" => "lemon", "key2" => "orange", "key3" => "banana");
asort($fruit);
print_r($fruit);
```

`arsort()` - Sorts an array in reverse order and maintains index association.

```
$fruit = array("key1" => "lemon", "key2" => "orange", "key3" => "banana");
arsort($fruit);
print_r($fruit);
```

`ksort()` - Sorts an array by key.

```
$fruit = array("d" => "lemon", "a" => "orange", "b" => "banana");
ksort($fruit);
print_r($fruit);
```

`krsort()` - Sorts an array by key in reverse order.

```
$fruit = array("d" => "lemon", "a" => "orange", "b" => "banana");
krsort($fruit);
print_r($fruit);
```

`natsort()` - Sorts an array using natural order algorithm.

```
$files = array("img1.png", "img10.png", "img12.png", "img2.png");
natsort($files);
print_r($files);
```

`natcasesort()` - Sorts an array using a case-insensitive natural order algorithm.

```
$files = array("Img1.png", "img10.png", "img12.png", "IMG2.png");
natcasesort($files);
print_r($files);
```

`uasort()` - Sorts an array with a user-defined comparison function and maintains index association.

```
$people = array( "Peter" => 35, "Jack" => 32,  "Mary" => 28);
function ageComparator($a, $b) {
```

```
    return $a - $b;
}
uasort($people , 'ageComparator');
print_r($people);
```

`uksort()` - Sorts an array by keys using a user-defined comparison function.

```
$people = array( "Peter" => 35, "Jack" => 32,  "Mary" => 28);
function ageComparator($a, $b) {
    return $a <=> $b;
}
uksort($people , 'ageComparator');
print_r($people);
```

#php