DATA STRUCTURES
ARRAYS

Andrew Sheehan

Boston University
Metropolitan College

INITIALIZATION
SYNTAX

```
<?php
  $names =
          array(“Andrew”, “Frank”);
?>
```

CREATION
DYNAMIC LENGTH

```
<?php
    $array[ ] = “Adding a value”;
?>
```

CREATION
GETTING A VALUE

```
<?php
   $val = $names[17];
?>
```

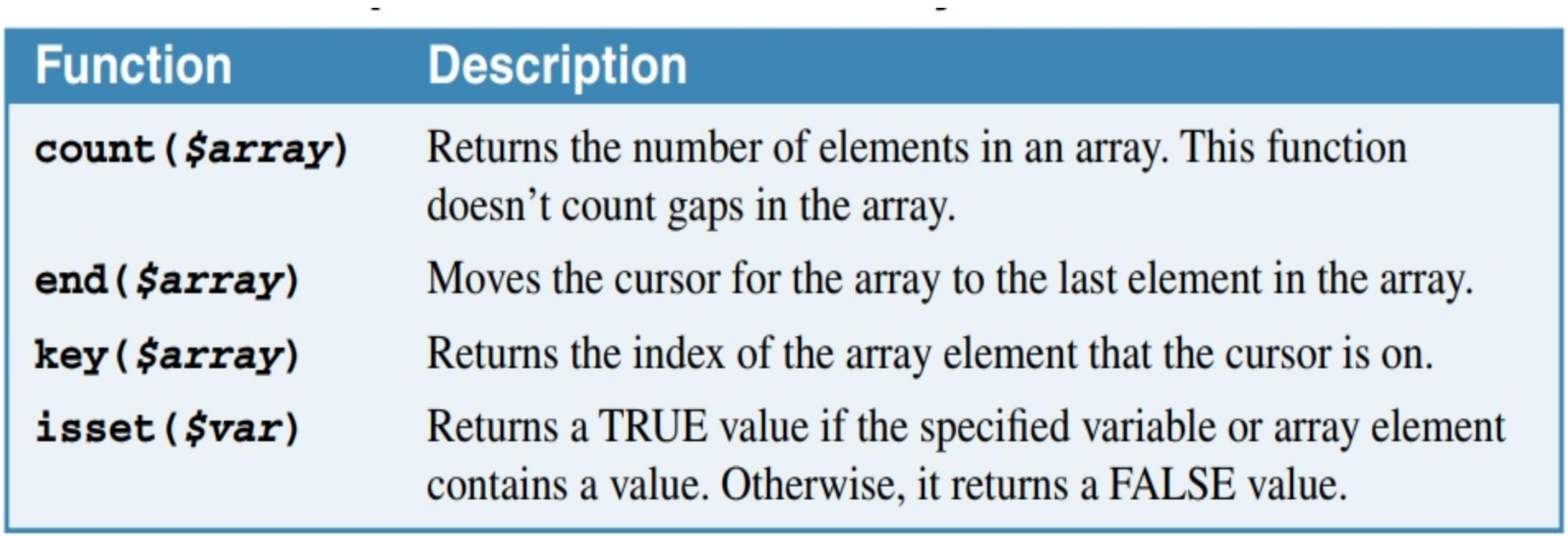
CREATION
REMOVING A VALUE

```
<?php
  // $array[17] will be NULL
  unset($array[17]);

  // $array will be NULL
  unset($array)
?>
```

PRINT_R()
USE/EXAMPLE

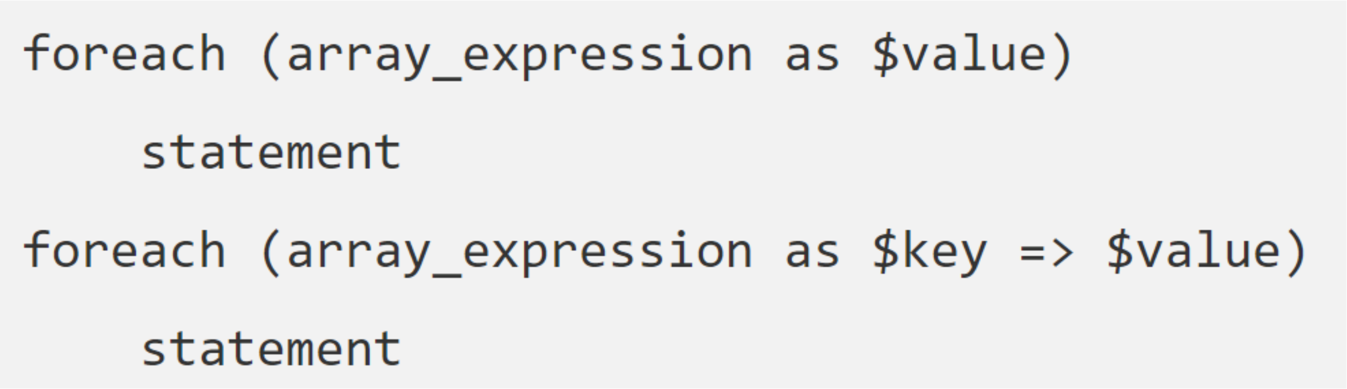
```
<?php
    // all indices and values
   print_r($names);
?>
```

BUILT-IN
ARRAY FUNCTIONS



Function	Description
`count($array)`	Returns the number of elements in an array. This function doesn’t count gaps in the array.
`end($array)`	Moves the cursor for the array to the last element in the array.
`key($array)`	Returns the index of the array element that the cursor is on.
`isset($var)`	Returns a TRUE value if the specified variable or array element contains a value. Otherwise, it returns a FALSE value.

BUILT-IN ITERATION



```
foreach (array_expression as $value)
    statement
foreach (array_expression as $key => $value)
    statement
```

```
<?php
$arr = array(1, 2, 3, 4);
foreach ($arr as &$value) {
    $value = $value * 2;
}
```

BUILT-IN
ARRAY ITERATION

```
foreach (array_expression as $value)
    statement
foreach (array_expression as $key => $value)
    statement
```

```php
<?php
$arr = array(1, 2, 3, 4);
foreach ($arr as &$value) {
    $value = $value * 2;
}
```

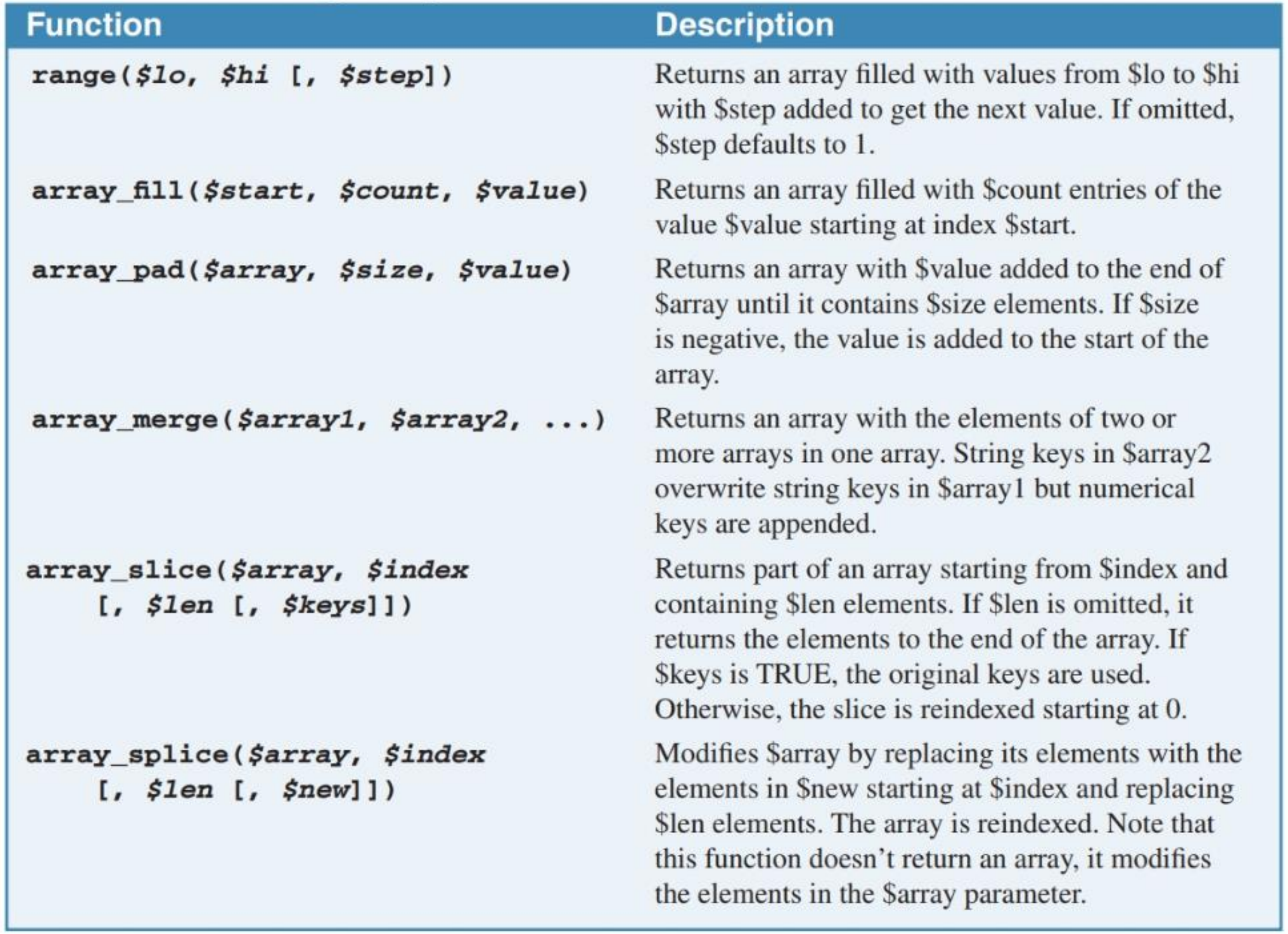
ASSOCIATIVE ARRAYS

```
<?php
/* First method to create an associate array. */
$student_one = array("Maths"=>95, "Physics"=>90,
                     "Chemistry"=>96, "English"=>93,
                     "Computer"=>98);

/* Second method to create an associate array. */
$student_two["Maths"] = 95;
$student_two["Physics"] = 90;
$student_two["Chemistry"] = 96;
$student_two["English"] = 93;
$student_two["Computer"] = 98;
?>
```

ASSOCIATIVE ARRAYS
ITERATING

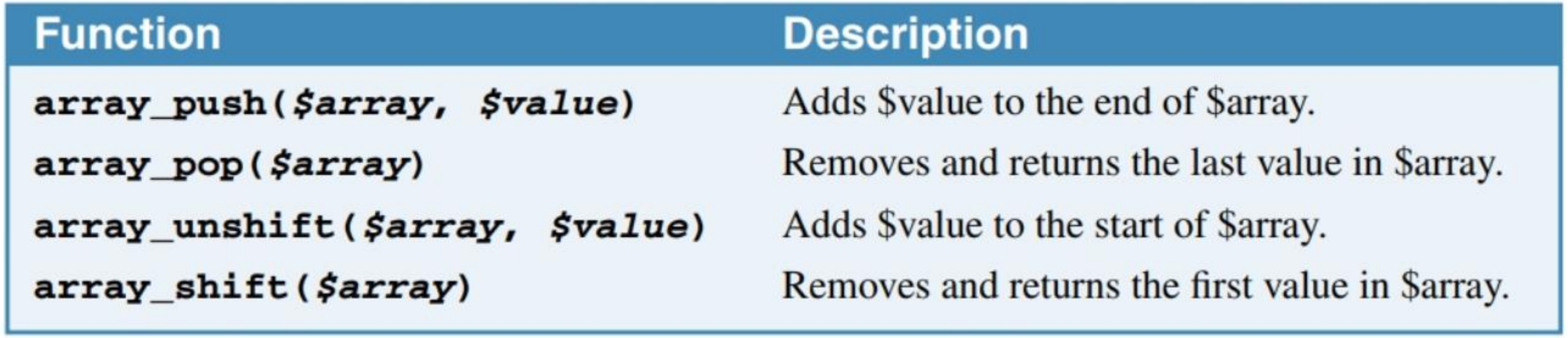
```php
<?php

/* Creating an associative array */
$student_one = array("Maths"=>95, "Physics"=>90,
                  "Chemistry"=>96, "English"=>93,
                  "Computer"=>98);


/* Looping through an array using foreach */
echo "Looping using foreach: \n";
foreach ($student_one as $subject => $marks){
    echo "Student one got ".$marks." in ".$subject."\n";
}

/* Looping through an array using for */
echo "\nLooping using for: \n";
$subject = array_keys($student_one);
$marks = count($student_one);

for($i=0; $i < $marks; ++$i) {
    echo $subject[$i] . ' ' . $student_one[$subject[$i]] . "\n";
}
?>
```

ARRAY FUNCTIONS (PARTIAL)



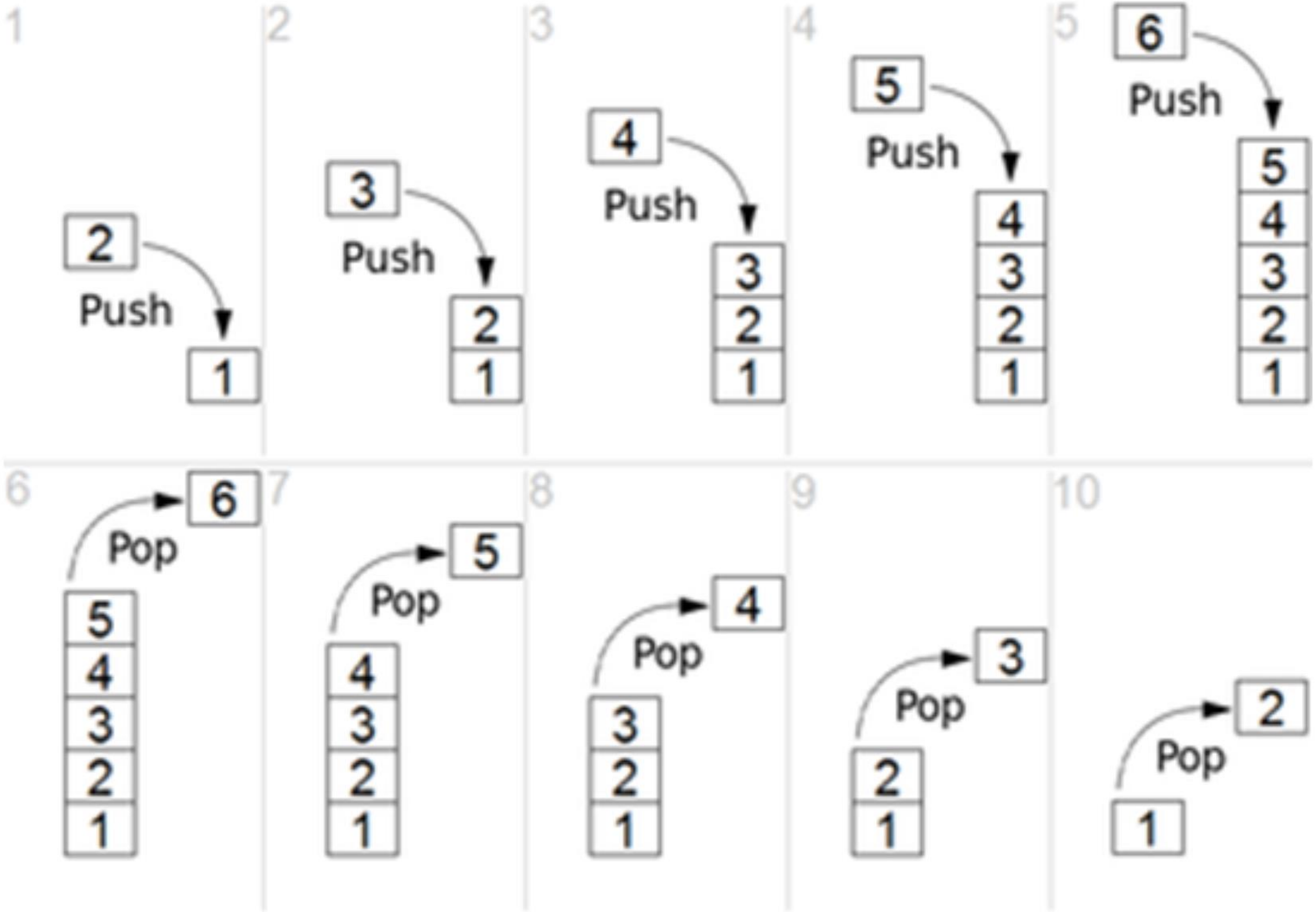
Function	Description
`range($lo, $hi [, $step])`	Returns an array filled with values from $lo to $hi with $step added to get the next value. If omitted, $step defaults to 1.
`array_fill($start, $count, $value)`	Returns an array filled with $count entries of the value $value starting at index $start.
`array_pad($array, $size, $value)`	Returns an array with $value added to the end of $array until it contains $size elements. If $size is negative, the value is added to the start of the array.
`array_merge($array1, $array2, ...)`	Returns an array with the elements of two or more arrays in one array. String keys in $array2 overwrite string keys in $array1 but numerical keys are appended.
`array_slice($array, $index [, $len [, $keys]])`	Returns part of an array starting from $index and containing $len elements. If $len is omitted, it returns the elements to the end of the array. If $keys is TRUE, the original keys are used. Otherwise, the slice is reindexed starting at 0.
`array_splice($array, $index [, $len [, $new]])`	Modifies $array by replacing its elements with the elements in $new starting at $index and replacing $len elements. The array is reindexed. Note that this function doesn't return an array, it modifies the elements in the $array parameter.

DATA STRUCTURES:
STACKS

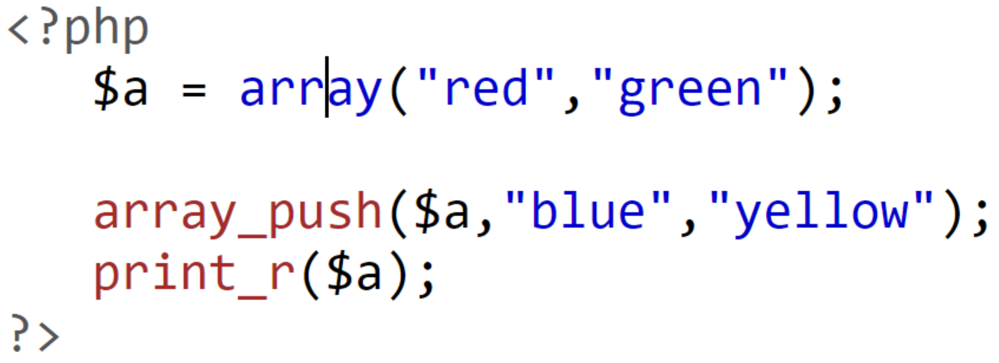


Function	Description
`array_push($array, $value)`	Adds $value to the end of $array.
`array_pop($array)`	Removes and returns the last value in $array.
`array_unshift($array, $value)`	Adds $value to the start of $array.
`array_shift($array)`	Removes and returns the first value in $array.

DATA STRUCTURES: STACKS



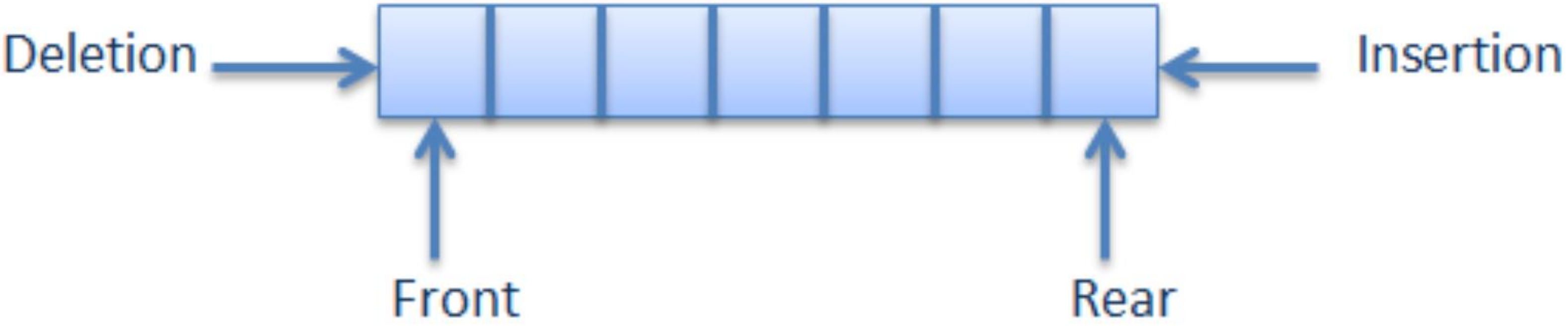
DATA STRUCTURES:
STACKS



```
<?php
   $a = array("red","green");

   array_push($a,"blue","yellow");
   print_r($a);
?>
```

DATA STRUCTURES: QUEUES

First In, First out


Deletion
Insertion
Front
Rear