Performance of array_shift and array_pop in PHP

We have confirmed that array_shift is much slower than array_pop in PHP.

Code:

```
<?

// Create an array with 100000 elements

$array = array();
for ($i = 0; $i < 100000; $i++) {
$array[] = rand();
}

// Remove the last 1000 elements using array_pop

$start = microtime(true);
for ($i = 0; $i < 1000; $i++) {
array_pop($array);
}
$stop = microtime(true);
printf("array_pop takes %.5f seconds\n", $stop – $start);

// Add back 1000 elements

for ($i = 0; $i < 1000; $i++) {
$array[] = rand();
}

// Remove the first 1000 elements using array_shift

$start = microtime(true);
for ($i = 0; $i < 1000; $i++) {
array_shift($array);
}
$stop = microtime(true);
printf("array_shift takes %.5f seconds\n", $stop – $start);

// Add back 1000 elements
```

```
for ($i = 0; $i < 1000; $i++) {
$array[] = rand();
}

// Remove the first 1000 elements by reversing the array and popping 1000 elements

$start = microtime(true);
$array_rev = array_reverse($array);
for ($i = 0; $i < 1000; $i++) {
array_pop($array_rev);
}
$stop = microtime(true);
printf("array_reverse + array_pop takes %.5f seconds\n", $stop – $start);

?>
```

Result:

```
array_pop takes 0.00089 seconds
array_shift takes 15.15544 seconds
array_reverse + array_pop takes 0.03934 seconds
```

Revision #1
Created 2008-07-28 15:44:06 UTC by Chan, Wing Kai
Updated 2008-07-28 15:44:06 UTC by Chan, Wing Kai