

Functional Programming with SPL Iterators

Debug Session

Marcus Börger

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```



```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
0 => 1
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
0 => 1
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
0 => 1
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```



```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
0 => 1
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
0 => 1
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
0 => 1
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
0 => 1
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
0 => 1
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
0 => 1
1 => 2
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
0 => 1
1 => 2
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
0 => 1
1 => 2
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```



```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
0 => 1
1 => 2
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
0 => 1
1 => 2
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
0 => 1
1 => 2
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
0 => 1
1 => 2
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
0 => 1
1 => 2
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
0 => 1
1 => 2
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
0 => 1
1 => 2
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
0 => 1
1 => 2
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```



```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
0 => 1
1 => 2
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
0 => 1
1 => 2
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
0 => 1
1 => 2
0 => 31
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
0 => 1
1 => 2
0 => 31
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
0 => 1
1 => 2
0 => 31
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
0 => 1
1 => 2
0 => 31
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
0 => 1
1 => 2
0 => 31
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
0 => 1
1 => 2
0 => 31
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```



```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
0 => 1
1 => 2
0 => 31
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
0 => 1
1 => 2
0 => 31
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
0 => 1
1 => 2
0 => 31
1 => 32
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
0 => 1
1 => 2
0 => 31
1 => 32
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
0 => 1
1 => 2
0 => 31
1 => 32
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
0 => 1
1 => 2
0 => 31
1 => 32
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
0 => 1
1 => 2
0 => 31
1 => 32
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
0 => 1
1 => 2
0 => 31
1 => 32
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```



```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
0 => 1
1 => 2
0 => 31
1 => 32
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
0 => 1
1 => 2
0 => 31
1 => 32
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
0 => 1
1 => 2
0 => 31
1 => 32
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
0 => 1
1 => 2
0 => 31
1 => 32
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
0 => 1
1 => 2
0 => 31
1 => 32
3 => 4
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
0 => 1
1 => 2
0 => 31
1 => 32
3 => 4
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
0 => 1
1 => 2
0 => 31
1 => 32
3 => 4
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
0 => 1
1 => 2
0 => 31
1 => 32
3 => 4
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```



```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
0 => 1
1 => 2
0 => 31
1 => 32
3 => 4
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```

```
<?php
$a = array('1', '2', array('31', '32'), '4');
$o = new RecursiveArrayIterator($a);
$i = new RecursiveIteratorIterator($o);
foreach($i as $key => $val) {
    echo "$key => $val\n";
}
?>
```

```
0 => 1
1 => 2
0 => 31
1 => 32
3 => 4
```

```
<?php
class RecursiveArrayIterator implements RecursiveIterator {
    protected $ar;
    function __construct(Array $ar) {
        $this->ar = $ar; }
    function rewind() {
        reset($this->ar); }
    function valid() {
        return !is_null(key($this->ar)); }
    function key() {
        return key($this->ar); }
    function current() {
        return current($this->ar); }
    function next() {
        next($this->ar); }
    function hasChildren() {
        return is_array(current($this->ar)); }
    function getChildren() {
        return new RecursiveArrayIterator(current($this->ar)); }
}
?>
```