### PHP 迭代器  http://blog.sina.com.cn/s/blog\_4d6c45250100xi30.html

说到迭代器这个东西，PHP内置的 迭代器还真多。迭代器到底来做什么的呢，其实就是用来遍历一个对象内部数据并且获得想要的结构。迭代器它可以控制foreach 语句的循环结构，通过适当的拓展可以获得指定情况下遍历的结果。就例如，PHP提供搜索，递归，聚集，XMl等各类型的迭代器。种类繁多，功能齐全，比如 要处理个XML，用个simplexmliterator 就立马见效。

下面介绍一下最基础的几个迭代器的接口：

Traversable : 这个是最初始的接口，他提供了控制foreach 语句的内置功能。所谓内置，就是只由PHP核心器去控制，外部无法调用。也就是说这个接口是无法被外部应用所应用。

Iterator ： 这个接口继承了Traversable ，并且包含了**控制foreach 的五个重要的函数 : rewind() , valid(), key(),current(),next()** ，它实现了foreach每一步的动作。

iteratoraggregate : 这个接口以继承了Traversable, 它最大的功能是可以将实现Iterator的类移除在外，自身不用去实现那五个方法便可实现控制foreach的目的。

<?php

class myIterator implements Iterator {

private $position = 0 ;

private $array = array(

"firstelement" ,

"secondelement" ,

"lastelement" ,

);

public function \_\_construct () {

$this -> position = 0 ;

}

function rewind () {

var\_dump ( \_\_METHOD\_\_ );

$this -> position = 0 ;

}

function current () {

var\_dump ( \_\_METHOD\_\_ );

return $this -> array [ $this -> position ];

}

function key () {

var\_dump ( \_\_METHOD\_\_ );

return $this -> position ;

}

function next () {

var\_dump ( \_\_METHOD\_\_ );

++ $this -> position ;

}

function valid () {

var\_dump ( \_\_METHOD\_\_ );

return isset( $this -> array [ $this -> position ]);

}

}

$it = new myIterator ;

foreach( $it as $key => $value ) {

var\_dump ( $key , $value );

}

执行结果:

