https://www.cnblogs.com/DeanChopper/p/4770572.html  
<https://www.cnblogs.com/whoknows/articles/adapter_in_php.html>

简单的来说适配器模式就是当你在使用新的接口去替代旧的接口去实现同一功能时，不需要改变原来的方法，只需要使用现有的接口去使用新的就好了。可以省去大量重复的代码。

## 适配器模式：

当引用的第三方类库或APi改变时不可能去改大量的代码一次产生了适配器模式

其实换成现实中的另一种说法：当你将程序完整的写结束时，你的客户突然说：听说pdo比mysql好，要不用pdo吧？客户笑眯眯的看着你，你傻乎乎的望着客户，哎，改吧！  
总不见得要所有代码都修改一遍吧，这时候适配器模式就派上了用场！

**<?php  
interface** toy{  
 **public function** openMouse();  
 **public function** closeMouse();  
}  
**class** dog **implements** toy{  
 **public function** openMouse()  
 {  
 **echo"小狗张嘴"**;  
 }  
  
 **public function** closeMouse()  
 {  
 **echo"小狗闭嘴"**;  
 }  
}  
  
  
*//目标接口***interface** target {  
 **public function** doopenMouse();  
 **public function** docloseMouse();  
}  
*//适配器类***class** adapter **implements** target{  
 **private $swe**;  
 **function** \_\_construct(toy $data){  
 $this->**swe**=$data;  
 }  
 *//委派$swe调用TOY的openMouse closeMouse方法* **public function** doopenMouse()  
 {  
 $this->**swe**->openMouse();  
 }  
  
 **public function** docloseMouse()  
 {  
 $this->**swe**->closeMouse();  
 }  
}  
$dog=**new** dog();  
$myadapter=**new** adapter($dog);  
$myadapter->doopenMouse();  
**?>**