# zval变量容器

每个php变量存在一个叫"zval"的变量容器中。

一个zval变量容器，除了包含变量的类型和值，还包括两个字节的额外信息。

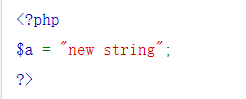
* 第一个是"is\_ref"，是个bool值，用来标识这个变量是否是属于引用集合(reference set)。通过这个字节，php引擎才能把普通变量和引用变量区分开来
* 由于php允许用户通过使用&来使用自定义引用，zval变量容器中还有一个内部引用计数机制，来优化内存使用。第二个额外字节是"refcount"，用以表示指向这个zval变量容器的变量(也称符号即symbol)个数。

所有的符号存在一个符号表中，其中每个符号都有作用域(scope)，那些主脚本(比如：通过浏览器请求的的脚本)和每个函数或者方法也都有作用域。

# 例子

## 生成一个新的zval容器

当一个变量被赋常量值时，就会生成一个zval变量容器，如下例这样：

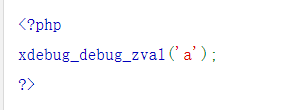


在上例中，新的变量a，是在当前作用域中生成的。

并且生成了类型为 string 和值为new string的变量容器。

在额外的两个字节信息中，"is\_ref"被默认设置为 false，因为没有任何自定义的引用生成。"refcount" 被设定为 1，因为这里只有一个变量使用这个变量容器.

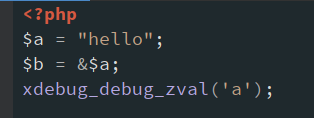
如果你已经安装了[» Xdebug](http://xdebug.org/" \o "Link : http://xdebug.org/)，你能通过调用函数 **xdebug\_debug\_zval()**显示"refcount"和"is\_ref"的值。



以上例程会输出：

a: (refcount=1, is\_ref=0)='new string'

## 增加一个zval的引用计数

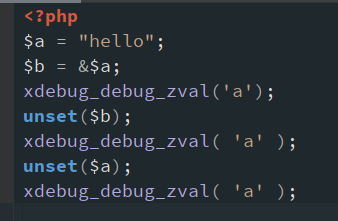


输出：

a: (refcount=2, is\_ref=1)='hello'

## 减少引用计数

变量容器在"refcount"变成0时就被销毁. 当任何关联到某个变量容器的变量离开它的作用域(比如：函数执行结束)，或者对变量调用了函数 [unset()](mk:@MSITStore:C:\\Users\\张卫航\\Desktop\\php_enhanced_zh.chm::/res/function.unset.html)时，"refcount"就会减1，下面的例子就能说明:



输出：

a: (refcount=2, is\_ref=1)='hello'

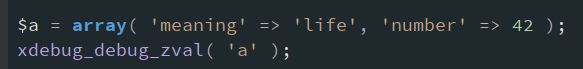
a: (refcount=1, is\_ref=1)='hello'

a: no such symbol （翻译：没有这样的符号）

## 复合类型(Compound Types)

当考虑像 array和object这样的复合类型时，事情就稍微有点复杂. 与 标量(scalar)类型的值不同，array和 object类型的变量把它们的成员或属性存在自己的符号表中。这意味着下面的例子将生成三个zval变量容器。

### 创建一个数组zval



输出：

a: (refcount=2, is\_ref=0)=array ('meaning' => (refcount=1, is\_ref=0)='life', 'number' => (refcount=0, is\_ref=0)=42)

这三个zval变量容器是: a，meaning和 number。增加和减少"refcount"的规则和上面提到的一样.

# 符号表是什么?

变量名字存哪去了?

符号表是指当前php页面中，所有变量名称的集合。

符号表相当于变量花名册。

符号表是一张[哈希表](https://so.csdn.net/so/search?q=%E5%93%88%E5%B8%8C%E8%A1%A8&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/u012164509/article/details/106257714/_blank),里面存储了变量名到变量的zval结构体的地址映射。

# 总结

即变量（符号）表管理所有变量名称。每个变量实际指向的是zval变量容器的地址，zval（Zend Value）变量容器是Zend Engine管理php变量的地方。

php没有变量的内存概念，拿不到Zend Engine的Zend Value的内存地址。

固，看不到00x....这种。