# Estructura de control

Estos incluyen if, por while, switch, etc. Este es un ejemplo if, ya que es el más complicado de ellos:

<? Php   
if (( condición1 ) || ( condition2 )) {   
    accion1 ;   
} elseif (( Condition3 ) && ( condition4 )) {   
    acción2 ;   
} else {   
    DefaultAction ;   
}   
?>

Las sentencias de control deben tener un espacio entre la palabra clave de control y paréntesis de apertura, para distinguirlas de las llamadas a funciones. Se le recomienda utilizar siempre llaves, incluso en situaciones en que son técnicamente opcional. Tenerlos aumenta la legibilidad y disminuye la probabilidad de errores de lógica que se introdujo cuando se añaden nuevas líneas.

<? Php   
interruptor ( condición ) {   
caso  1 :   
    accion1 ;   
    break; caso    
  
2 :   
    acción2 ;   
    break; por defecto:   
  
  
    DefaultAction ;   
    break;   
}   
?>

# Operadores alternarios

La misma regla que para if cláusulas también se aplica para el operador ternario : Puede dividirse en varias líneas, manteniendo el signo de interrogación y los dos puntos en la parte delantera.

<?php $a    
  
=  $condition1  &&  $condition2   
    ?  $foo  :  $bar ;   
  
$b  =  $condition3  &&  $condition4   
    ?  $foo\_man\_this\_is\_too\_long\_what\_should\_i\_do   
    :  $bar ;   
?>

# Definición de clases

Declaraciones de clase tienen su llave de apertura de una nueva línea:

<? Php   
clase  foo\_bar   
{   
  
    // ... código va aquí   
  
}   
?>

# Código PHP Etiquetas

Siempre use <? php?> código PHP para delimitar, no el <? ?> taquigrafía. Esto es necesario para el cumplimiento de PEAR y es también la forma más portátil para incluir código PHP en diferentes sistemas operativos y configuraciones.

# Comentarios

Bloques de documentación comentario en línea completos (docblocks) deben ser proporcionados. Por favor, lea el archivo de ejemplo y bloques de encabezado Comentar secciones de los estándares de codificación para aprender los detalles específicos de la escritura docblocks paquetes PEAR. Más información se puede encontrar en la phpDocumentor sitio web.

Comentarios no de documentación se recomienda encarecidamente. Una regla general es que si nos fijamos en una sección de código y pensar " Wow, yo no quiero tratar de describir ese ", es necesario comentar que antes se le olvida cómo funciona.

Comentarios de estilo C (/ \* \* /) y comentarios estándar C ++ (//) son ambos muy bien. El uso de comentarios de estilo Perl / shell (#) no se recomienda.

# Convenciones de nomenclatura

## Variables y Funciones Globales

Los nombres deben comenzar con un solo guión bajo seguido por el nombre del paquete y otro guión bajo.Por ejemplo, el paquete de PEAR usa una variable global llamada $ \_PEAR\_destructor\_object\_list.

Funciones globales deben ser nombrados a través del " tapas studly estilo "(también conocido como" caso lleno de baches "o" tapas de camellos "). Además, deben tener el nombre del paquete como un prefijo, para evitar colisiones de nombres entre los paquetes. La letra inicial del nombre (después del prefijo) es minúscula, y cada letra que comienza una nueva " palabra "se escribe con mayúscula. Un ejemplo:

## Clases

Las clases deben tener nombres descriptivos

Log Net\_Finger HTML\_Upload\_Error