

✓ Errores de Sintaxis/Programación: Corregidos

1. En el switch, se utiliza ``$form.len()`` en lugar de ``count($form)`` para obtener la longitud del array ``$form``.
2. En el bloque 2, en la condición del if, ``$form[2] = ""`` debería ser ``$form[2] == ""``.
3. En el bloque 2, en la asignación de ``$res``, se usa la sintaxis de objeto ``{}`` en lugar de corchetes ``[]``.
4. En el bloque 2, el caso ``$form.len()`` debería ser ``$form[0]``.
5. En el bloque 2, ``$form[2].$res`` debería ser ``$form[1].$res``.
6. En el bloque 3, la condición ``$c[1] != "0" and $form[0] != "0"`` debería ser ``$c[1] != "0" && $form[0] != "0"``.
7. En el bloque 3, ``$res.="0";`` debería ser ``$res .= "0";``.

Bloque 1:

Este bloque divide la parte entera y decimal del número de entrada. Si el número es negativo, cambia la parte entera a "-0". En el caso de ``$input == "-.40"``, el valor de ``$c`` sería ``array("-0", "40")``.

Bloque 2:

Este bloque formatea la parte entera del número según las especificaciones dadas en el array ``$form``. En el caso de ``$form = array("1", ",", ";")`` y ``$c = array("-3000", "20")``, el valor de ``$res`` sería ``"-3,000.20"``.

Bloque 3:

Este bloque agrega la parte decimal al resultado si es diferente de "0" y si el formato especifica una cantidad de dígitos decimales diferentes de "0". En el caso de ``$res = "20"``, ``$form = array("1", ",", ";")`` y ``$c = array("-3000", "20")``, el valor final de ``$res`` sería ``"-3,000.20"``.

Función convertir:

La función `convertir` toma un número de entrada y un formato opcional y devuelve el número formateado según las especificaciones dadas en el formato. La función realiza principalmente operaciones de formateo en la parte entera y decimal del número.

Funciones similares en PHP:

PHP tiene varias funciones incorporadas que pueden realizar tareas similares, como `number_format` para el formateo de números y `str_replace` para reemplazar caracteres en una cadena. Estas funciones pueden simplificar y mejorar la legibilidad del código. Ext

Código fuente corregido

```
<?php
```

```
function convertir($input, $formato = "") {  
    global $main_system;  
  
    $res = $input;  
  
    if ($input !== "") {  
        if ($formato === "" && isset($main_system['formato'])) {  
            $formato = $main_system['formato'];  
        }  
  
        if ($formato !== "") {  
            $form = explode("|", $formato);  
            for ($i = 0; $i < count($form); $i++) {  
                $form[$i] = trim($form[$i]);  
            }  
  
            switch (count($form)) {
```

case 1:

```
$form[1] = "."; // Corrección: Añadir punto en el caso 1
```

case 2:

```
$form[2] = ""; // Corrección: Asignar cadena vacía en el caso 2
```

case 3:

```
// BLOQUE 1
```

```
$c = array("0", "0");
```

```
$r = strpos($input, ".");
```

```
if ($r !== false) {
```

```
    $c[0] = substr($input, 0, $r);
```

```
    $c[1] = substr($input, $r + 1);
```

```
} else {
```

```
    $c[0] = $input;
```

```
}
```

```
if ($c[0] === "-") {
```

```
    $c[0] = "-0";
```

```
}
```

```
// FIN BLOQUE 1
```

```
$res = "";
```

```
// BLOQUE 2
```

```
if ($form[2] == "") {
```

```
    $res = $c[0];
```

```
} else {
```

```
    $l = 0;
```

```
    if (substr($c[0], 0, 1) == "-") {
```

```
        $l = 1;
```

```

    }

    for ($j = strlen($c[0]) - 1, $z = 0; $j >= $l; $j--, $z++) {
        if ($z > 0 && $z % 3 == 0) {
            $res = $form[1] . $res; // Corrección: Utilizar $form[1] en lugar de
$form[2]
        }

        $res = substr($c[0], $j, 1) . $res;
    }

    if (substr($c[0], 0, 1) == "-") {
        $res = "-" . $res;
    }
}

// FIN BLOQUE 2

// BLOQUE 3

if ($c[1] != "0" && $form[0] != "0") {
    $res .= $form[1];
    for ($j = 0; $j < $form[0]; $j++) {
        if ($j < strlen($c[1])) {
            $res .= $c[1][$j];
        } else {
            $res .= "0";
        }
    }
}

if ($res === "-0") {
    $res = "0";
}

```

```
// FIN BLOQUE 3
```

```
break; // Corrección: Agregar break al final del switch
```

```
}
```

```
}
```

```
}
```

```
exit($res);
```

```
}
```

```
?>
```