



# Introduction to PHP

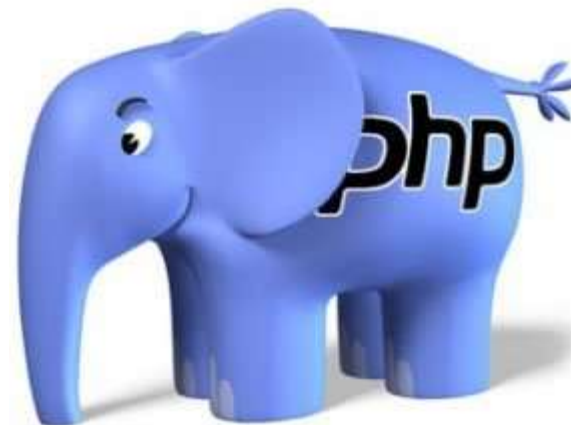




## 6. Manejo de errores.



- Manejo y tipos de errores.
- Manejo de excepciones.
- Uso de break y continue.
- Uso de punteros (referencias).



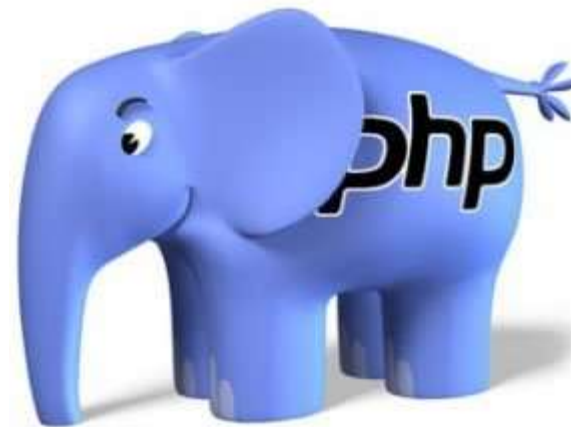
# Manejo de errores.

Estas funciones se ocupan del manejo y **registro de errores**.

Le permiten definir sus **propias reglas** de manejo de errores, así como modificar la manera en que los errores pueden ser registrados.

Esto le permite cambiar y mejorar la **notificación de errores** para acomodarla a sus necesidades.





# Tipos de errores.

Las funciones de notificación de errores le permiten personalizar qué niveles y tipos de errores se dan, abarcando desde simples avisos hasta funciones personalizadas devueltas al originarse un error.

## -> Errores más comunes:

1. Fatal error
2. Warning
3. Parse error.





## Fatal error.

*Aborta la ejecución del programa. Ocurre por ejemplo cuando mandamos a llamar una función que no existe.*

| (!) Fatal error: Uncaught Error: Call to undefined function sumar() in C:\wamp64\www\pruebas_web\errores.php on line 7 |        |        |          |                   |
|--|--------|--------|----------|-------------------|
| (!) Error: Call to undefined function sumar() in C:\wamp64\www\pruebas_web\errores.php on line 7                       |        |        |          |                   |
| Call Stack   |        |        |          |                   |
| #  | Time   | Memory | Function | Location          |
| 1  | 0.0005 | 364080 | {main}() | ...\errores.php:0 |





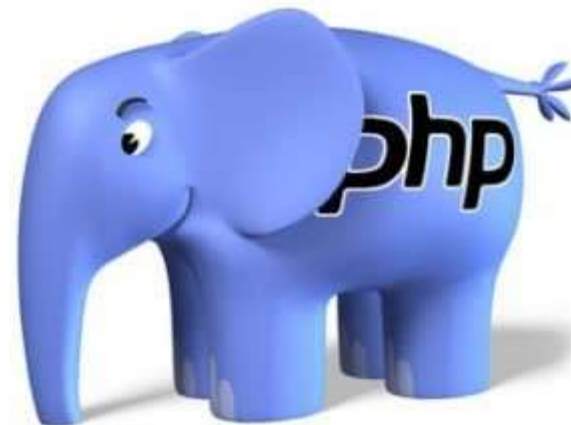
## Warning.

*No abortan la ejecución del programa, sólo nos avisa que algo podría mejorar, suceden por ejemplo cuando dividimos una variable por cero.*

| (!) Warning: Division by zero in C:\wamp64\www\pruebas_web\errores.php on line 8 |        |        |          |                   |
|--|--------|--------|----------|-------------------|
| Call Stack   |        |        |          |                   |
| #  | Time   | Memory | Function | Location          |
| 1  | 0.0005 | 363024 | {main}() | ...\errores.php:0 |

INF





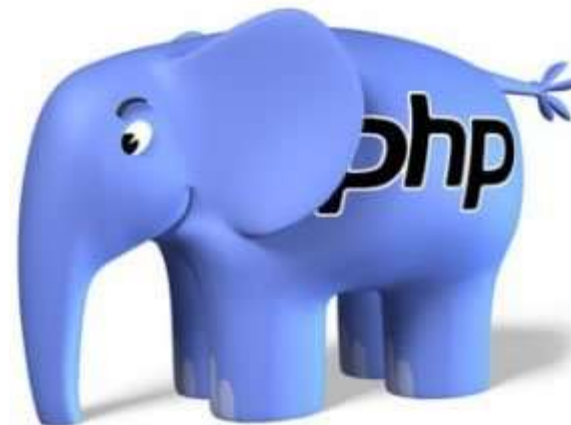
## Parse error.

*Son errores de tipo análisis gramatical, aparecen cuando tenemos errores de sintaxis. Se aborta la ejecución del programa. Ejemplo: olvidar un punto y coma.*

**(!)** Parse error: syntax error, unexpected 'echo' (T\_ECHO) in C:\wamp64\www\pruebas\_web\errores.php on line 8







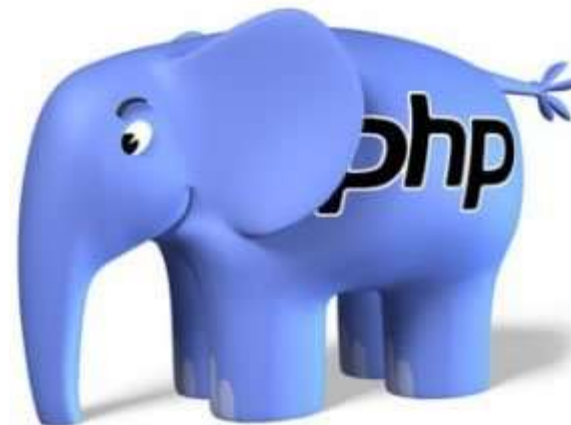
# Manejo de excepciones.

Una excepción puede ser lanzada ("throw"), y atrapada ("catch") dentro de PHP. El código puede estar dentro de un bloque try para facilitar la captura de excepciones potenciales.

```
try {  
    $x=(int)readline(" Valor de x: ");  
    if ($x===0) {  
        throw new Exception('División por cero.');    }else echo "1/$x";  
} catch (Exception $e) {  
    echo 'Excepción capturada: ', $e->getMessage(), "\n";  
}finally { echo "Lanzando un final";}
```





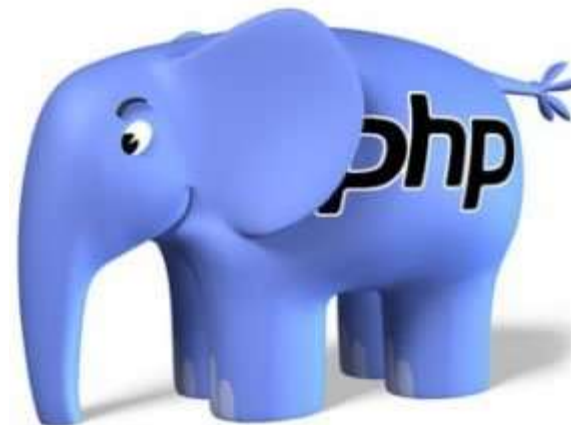


## Break y continue.

**break** finaliza la ejecución de la estructura for, foreach, while o switch en curso.

**continue** se utiliza dentro de las estructuras iterativas para saltar el resto de la iteración actual del bucle y continuar la ejecución en la evaluación de la condición, para luego comenzar la siguiente iteración.





# Referencias en PHP.

Las Referencias en PHP son medios de acceder al mismo contenido de una variable mediante diferentes nombres. No son como los punteros de C; por ejemplo, no se puede realizar aritmética de punteros con ellas, realmente no son direcciones de memoria, etc.

```
<?php  
$var = "Hola";  
$ref =&$var; // nuevo objeto que referencia a $var  
?>
```

