

Documentación

DOCUMENTACIÓN

Un comentario es una anotación en el código que el compilador o interprete va a ignorar

DOCUMENTACIÓN

- Los comentarios sirven para explicar el *objetivo de las sentencias* y también para explicar *qué realiza un método* o *qué implementa una clase* (no cómo lo hace).
- Si modificamos el código también debemos modificar los comentarios

DOCUMENTACIÓN

OBJETIVO

EXPLICAR POR QUÉ SE HACE

- FINALIDAD DE UNA CLASE
- QUÉ HACE UN MÉTODO
- PARA QUÉ SIRVE UNA VARIABLE
- QUÉ ALGORITMO SE USA
- QUÉ SE PODRÍA MEJORAR

IMPORTANCIA

- Fundamental para detectar errores.
- Posterior mantenimiento

PRÓPOSITOS DE LOS COMENTARIOS

- **Explicar el objetivo de las sentencias** .- con el objeto de conocer la función de esa sentencia.
- Explicar **qué realiza** un método, o clase, *no cómo lo realiza* . Valores que devuelve un método

COMENTARIOS

// seguidos del comentario

/ COMENTARIO */*

```
<?php
    echo 'Esto es una prueba'; // Esto es un comentario al estilo de c++ de una sola línea
    /* Esto es un comentario multilínea
       y otra línea de comentarios */
    echo 'Esto es otra prueba';
    echo 'Una prueba final'; # Esto es un comentario al estilo de consola de una sola línea
?>
```

DOCBLOCK

```
<?php  
/** This is a single line DocComment*/  
  
/**  
 * This is a multi-line DocComent  
 */  
function associatedFunction()  
{  
  
}
```

PHPDoc

- PhpDocumentor2 es una herramienta con la cual es posible generar documentación desde su código fuente PHP.
- Proporcionar a los desarrolladores de PHP con todas las herramientas y recursos necesarios para escribir documentación eficaz y completa con el menor esfuerzo posible.

PARTES DOCBLOCK

Sumario (Summary) Breve descripción de la función básica de un elemento.

Descripción (Description) Descripción larga. Una descripción no debe existir sin sumario.

Tags y anotaciones Nos permite añadir metainformación, cada tag precede @ y por línea.

DOCBLOCK

```
<?php
```

```
/**
```

- * A **summary** informing the user what the associated element does.

- *

- * A ***description***, that can span multiple lines, to go *_in-depth_* into the details of this element

- * and to provide some background information or textual references.

- *

- * **@param string \$myArgument** With a **description** of this argument, these may also

- * span multiple lines.

- *

- * **@return** void

- */

```
function myFunction($myArgument)
```

TAG

@api

@author

@category

@copyright

@deprecated

@example

@filesource

@global

@ignore

@internal

@license

@link

@method

@package

@param

@property

@property-read

@property-write

@return

@see

@since

@source

@subpackage

@throws

@var

@version

PHPDOCUMENTOR

- Herramienta para generar código de forma automática a partir del código fuente.
- Nos permite definir código en cada clase, cada método y cada función de nuestro código.

PHPDOCUMENTOR: GENERAR DOCUMENTACIÓN

- Desde línea de comandos (php CLI-Command Line Interpreter)
- Desde interfaz web (incluida en la distribución)
- Desde código. Como phpDocumentor está desarrollada en PHP, puede incluir su funcionalidad dentro de scripts propios.

PHPDOCUMENTOR: FORMATOS SALIDA

- 1) HTML a través de un buen número de plantillas predefinidas.
- 2) PDF
- 3) XML (DocBook) Podemos transformar XSLT a cualquier otro utilizando nuestras propias reglas y hojas de estilos.