

<?php

BASES DE
php

IMPARTIDO POR
JOSE IVAN SUAZO



<https://github.com/netsti/Taller-PHP>

?>

¿COMO INICIO *php*?



Rasmus Lerdorf

desarrollo un código que le ayudaría a crear su página web personal de manera más sencilla

PHP Tools

Personal Home Page Tools o herramientas para páginas iniciales personales

v1

v3

PHP Hypertext Preprocessor

Deciden que el nombre ya le quedaba un poco corto al lenguaje y deciden cambiar el significado de las siglas

<?php

SINTAXIS

DELIMITADOR

<?php

El intérprete de PHP solo ejecuta el código que se encuentra entre sus delimitadores.

Los delimitadores más comunes son ***<?php*** para abrir una sección PHP y ***?>*** para cerrarla.

...

?>

El propósito de estos delimitadores es separar el código PHP del resto de código, como por ejemplo el HTML

SINTAXIS

VARIABLES

`$variable = valor;`

prefijo ———— *final de
sentencia*

Las variables se prefijan con el símbolo del dólar (\$) y no es necesario indicar su tipo. Las variables, a diferencia de las funciones, distinguen entre mayúsculas y minúsculas.

Las cadenas de caracteres pueden ser encapsuladas tanto en dobles comillas como en comillas simples, aunque en el caso de las primeras, se pueden insertar variables en la cadena directamente, sin necesidad de concatenación.

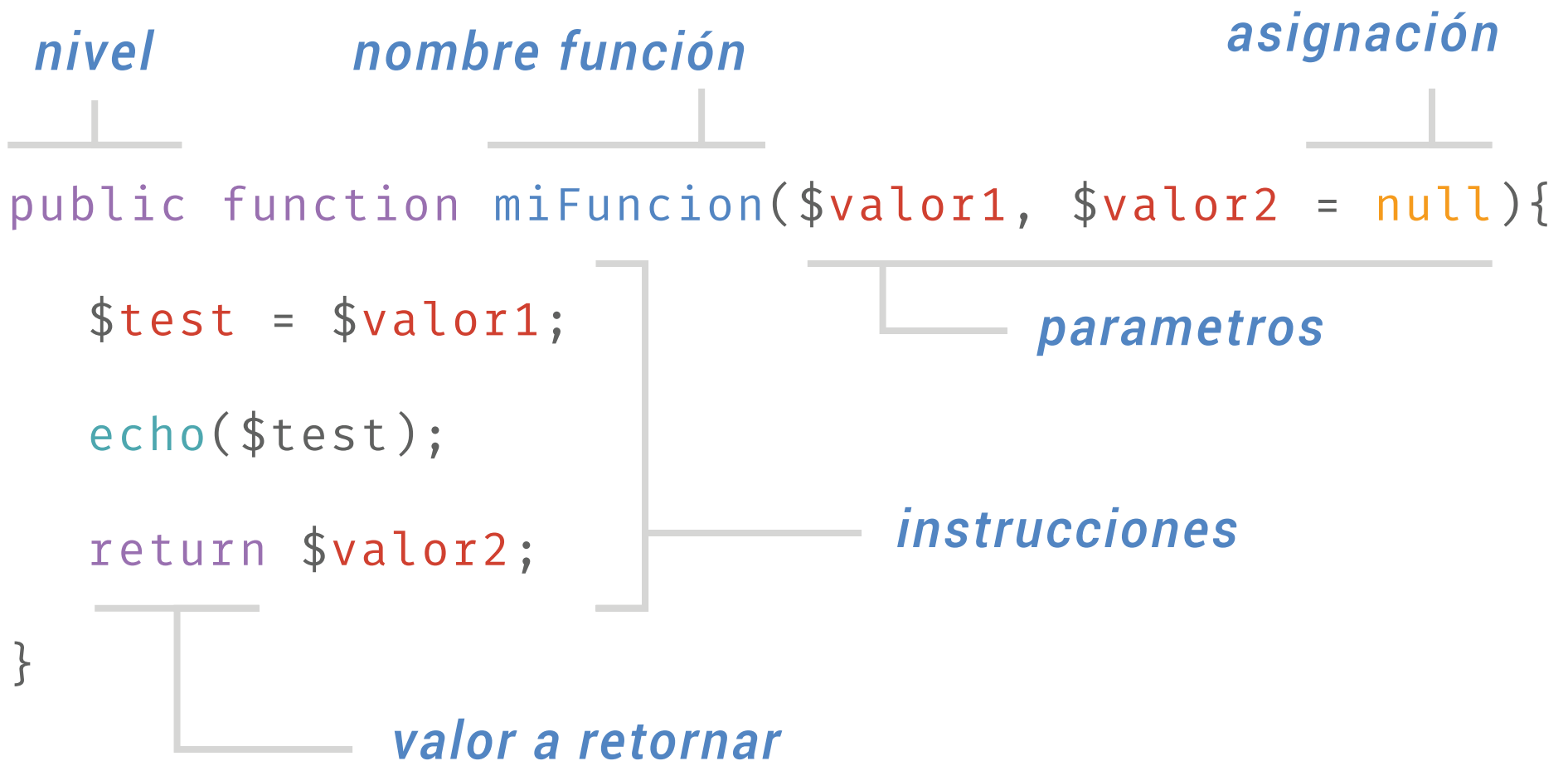
SINTAXIS

COMENTARIOS

```
// comentario de una linea  
/* este es un comentario  
de varias lineas  
*/
```

Los comentarios se pueden escribir bien con dos barras al principio de la línea, o con una almohadilla. También permite comentarios multi-línea encapsulados en `/* */`

SINTAXIS **METODOS**



The diagram illustrates the syntax of a PHP function with the following components and labels:

- nivel**: Points to the `public` keyword.
- nombre función**: Points to the `function` keyword.
- asignación**: Points to the `miFuncion` name.
- parametros**: Points to the parameter list `($valor1, $valor2 = null)`.
- instrucciones**: Points to the block of code inside the function: `$test = $valor1;`, `echo($test);`, and `return $valor2;`.
- valor a retornar**: Points to the `return` statement and its value `$valor2`.

```
public function miFuncion($valor1, $valor2 = null){  
    $test = $valor1;  
    echo($test);  
    return $valor2;  
}
```

SINTAXIS

SUPERGLOBALES

`$GLOBALS` Hace referencia a todas las variables disponibles en el ámbito global

```
<?php
function test() {
    $foo = "variable local";

    echo '$foo en el ámbito global: ' . $GLOBALS["foo"] . "\n";
    echo '$foo en el ámbito simple: ' . $foo . "\n";
}
```

```
$foo = "Contenido de ejemplo";
test();
?>
```

```
$foo en el ámbito global: Contenido de ejemplo
$foo en el ámbito simple: variable local
```

SINTAXIS

SUPERGLOBALES

`$_SERVER` Información del entorno del servidor y de ejecución

```
<?php
```

```
    echo 'IP Servidor: ' . $_SERVER["SERVER_ADDR"];  
    echo 'Nombre Servidor: ' . $_SERVER["SERVER_NAME"];  
    echo 'Software Servidor: ' . $_SERVER["SERVER_SOFTWARE"];  
    echo 'IP Remota: ' . $_SERVER["REMOTE_ADDR"];  
    echo 'Puerto Remoto: ' . $_SERVER["REMOTE_PORT"];
```

```
?>
```


SINTAXIS

SUPERGLOBALES

`$_GET` & `$_POST` Un array asociativo de variables pasado al script actual vía parámetros URL

```
<?php
function printVars($vars) {
    foreach($vars as $key => $value){
        echo "$key = $value; \n";
    }
}
```

```
printVars($_GET);
printVars($_POST);
?>
```

<?php

SINTAXIS

SUPERGLOBALES

\$_FILES

\$_REQUEST

\$_COOKIE

\$_ENV

\$_SESSION

SINTAXIS ***FUNCIONES ESPECIALES***

| | | |
|-------------------------|---------------------------|-----------------------------|
| <code>echo()</code> | <code>strtolower()</code> | <code>str_replace()</code> |
| <code>print()</code> | <code>strtoupper()</code> | <code>explode()</code> |
| <code>var_dump()</code> | <code>ucfirst()</code> | <code>implode()</code> |
| <code>count()</code> | <code>ucwords()</code> | <code>array_diff()</code> |
| <code>date()</code> | <code>trim()</code> | <code>array_serach()</code> |
| <code>mail()</code> | <code>isset()</code> | <code>in_array()</code> |
| <code>include()</code> | | |

PHP POO

CLASES

La definición básica de una clase comienza con la palabra reservada `class`, seguido de un nombre de clase, y continuando con un par de llaves que encierran las definiciones de las propiedades y métodos pertenecientes a dicha clase.

```
<?php namespace Yo\Ejemplo;
use Autor\Modulo\Clase;

class ClaseSencilla extend Clase{
    //Declaración de una propiedad
    public $var = 'un valor predeterminado';

    // Declaración de un método
    public function mostrarVar() {
        echo $this->var;
    }
}
?>
```

PHP POO ***PROPIEDADES***

Las variables pertenecientes a una clase se llaman "propiedades". También se les puede llamar usando otros términos como "atributos" o "campos", pero para los propósitos de esta referencia se va a utilizar "propiedades". Éstas se definen usando una de las palabras reservadas `public`, `protected`, o `private`, seguido de una declaración normal de variable.

Esta declaración puede incluir una inicialización, pero esta inicialización debe ser un valor constante, es decir, debe poder ser evaluada durante la compilación y no depender de información generada durante la ejecución.

PHP POO

PROPIEDADES

```
<?php
class ClaseSencilla{
    //Declaración de propiedades inválidas:
    public $var1 = 'hola'.'mundo';
    public $var2 = 1+2;
    public $var3 = self::MiMetodoEstatico();
    public $var4 = $otraVariable;

    // Declaración de propiedades válidas:
    public $var5 = MICONSTANTE;
    public $var6 = array(true, false);
}
?>
```

PHP POO

CONSTRUCTORES

```
<?php
class BaseClass{
    function __construct($initialVar = true){
        $this->var = $initialVar;
        echo "En el constructor BaseClass\n";
    }
}

class SubClass extend BaseClass{
    function __construct() {
        parent::__construct();
        echo "En el constructor SubClass\n";
    }
}

?>
```