

DESARROLLO WEB CON PHP

Centro ADM

Barragán Jimenez Jonathan

23 de junio de 2018

Índice

- 1 Primer acercamiento
 - Aplicaciones Web
 - ¿Qué es PHP?

- 2 PHP
 - Tipos de datos
 - Estructuras de control de flujo
 - funciones

Índice

1 Primer acercamiento

- Aplicaciones Web
- ¿Qué es PHP?

2 PHP

- Tipos de datos
- Estructuras de control de flujo
- funciones

Índice

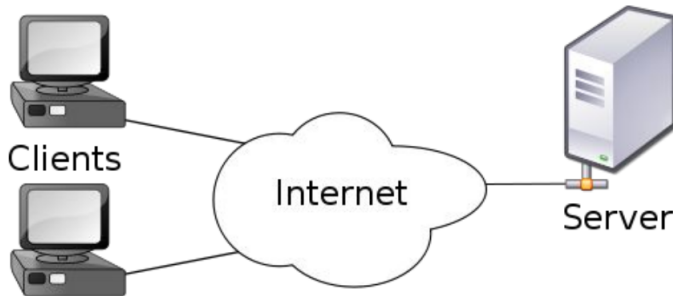
1 Primer acercamiento

- Aplicaciones Web
- ¿Qué es PHP?

2 PHP

- Tipos de datos
- Estructuras de control de flujo
- funciones

¿Cómo funcionan?



Índice

1 Primer acercamiento

- Aplicaciones Web
- ¿Qué es PHP?

2 PHP

- Tipos de datos
- Estructuras de control de flujo
- funciones

PHP

- PHP: Hypertext Preprocessor

PHP

- PHP: Hypertext Preprocessor
- Lenguaje de programación multiparadigma (no siempre es bueno que entre más mejor)

PHP

- PHP: Hypertext Preprocessor
- Lenguaje de programación multiparadigma (no siempre es bueno que entre más mejor)
- Interpretado (Scripting Language)

PHP

- PHP: Hypertext Preprocessor
- Lenguaje de programación multiparadigma (no siempre es bueno que entre más mejor)
- Interpretado (Scripting Language)
- Trabaja del lado del servidor

PHP

- PHP: Hypertext Preprocessor
- Lenguaje de programación multiparadigma (no siempre es bueno que entre más mejor)
- Interpretado (Scripting Language)
- Trabaja del lado del servidor
- Ayuda a hacer páginas web dinámicas

PHP

- PHP: Hypertext Preprocessor
- Lenguaje de programación multiparadigma (no siempre es bueno que entre más mejor)
- Interpretado (Scripting Language)
- Trabaja del lado del servidor
- Ayuda a hacer páginas web dinámicas
- Utilizaremos la versión 7

INSTALACION: PHP Y GIT

Linux

- `sudo apt-get update`

INSTALACION: PHP Y GIT

Linux

- `sudo apt-get update`
- `sudo apt-get install php7.0`

INSTALACION: PHP Y GIT

Windows

- Instalar Microsoft's Web Platform Installer.

`https:`

`//www.microsoft.com/web/downloads/platform.aspx`

Windows

Instalar Git.

`https://git-scm.com/`

INSTALACION: PHP Y GIT

Windows

- Instalar Microsoft's Web Plataform Installer.
`https://www.microsoft.com/web/downloads/platform.aspx`
- Instalar php 7 (x64, si su compu es reciente) con WebPI

Windows

Instalar Git.

`https://git-scm.com/`

PRIMEROS PASO

- Mi primer script

PRIMEROS PASO

- Mi primer script
- EL interprete de php
\$ php <nombreArchivo >.php

PRIMEROS PASO

- Mi primer script
- EL interprete de php
\$ php <nombreArchivo >.php
- El verificador de sintaxis
\$ php -l <nombreArchivo >.php

PRIMEROS PASO

- Mi primer script
- EL interprete de php
\$ php <nombreArchivo >.php
- El verificador de sintaxis
\$ php -l <nombreArchivo >.php
- EL servidor in-Built
\$ php -S localhost:8000

PRIMEROS PASO

- Mi primer script
- EL interprete de php
\$ php <nombreArchivo >.php
- El verificador de sintaxis
\$ php -l <nombreArchivo >.php
- EL servidor in-Built
\$ php -S localhost:8000
- PHP y navegadores
url: localhost:8000/<nombreArchivo>.php

Índice

1 Primer acercamiento

- Aplicaciones Web
- ¿Qué es PHP?

2 PHP

- Tipos de datos
- Estructuras de control de flujo
- funciones

Índice

1 Primer acercamiento

- Aplicaciones Web
- ¿Qué es PHP?

2 PHP

- Tipos de datos
- Estructuras de control de flujo
- funciones

Tipos de datos

Versiones anteriores a la 7

- callable
- array
- object

En la versión 7

Tipos de datos

Versiones anteriores a la 7

- callable
- array
- object

En la versión 7

- string
- int
- float
- bool

Tipos de datos

Arreglos

Hay dos tipos de arreglos:

- 1 Arreglos con índices
`$a = ['uno', 'dos', 'tres'];`
`echo $a[3];`

Tipos de datos

Arreglos

Hay dos tipos de arreglos:

① Arreglos con índices

```
$a = ['uno', 'dos', 'tres'];  
echo $a[3];
```

② Arreglos con llaves (asociativos)

```
$b = ['1' => 'Lunes', '2' => 'Martes', '3' => 'Miercoles'];  
echo $b['3'];
```

Las llaves son cadenas y los valores de cualquier tipo.

Tipos de datos

Cadenas

Hay 4 maneras de definir una cadena, nos enfocaremos en 2

- ① Comillas simples ' ';
- ② Comillas dobles " ";

Los nombres de variables serán extendidos.

Tipos de datos

Cadenas

Hay 4 maneras de definir una cadena, nos enfocaremos en 2

- ① Comillas simples ' ';
- ② Comillas dobles " ";

Los nombres de variables serán extendidos.

Ejemplo

```
$var = "mundo";  
echo "hola $var";  
echo 'hola $var';
```

Tipos de datos

Booleanos

Los siguientes valores se pueden interpretar como falso

- FALSE
- int 0
- float 0.0
- string "" ó '0'
- array []
- NULL

Operadores de comparación

- Equivalencia (==):
devuelve verdadero si dos variables tienen valores iguales pero para compararlas fue necesario hacer algún *cast*

Operadores de comparación

- Equivalencia (==): devuelve verdadero si dos variables tienen valores iguales pero para compararlas fue necesario hacer algún *cast*
- Identidad (===): devuelve verdadero si las variables son del mismo tipo y además uno a uno sus valores son iguales. Devuelve falso en otro caso.

Tipos de datos

Las validaciones de los tipos se pueden dar de las siguientes dos maneras:

- Coercitiva (por omisión)
 - Hace castings implícitos. Se forza el valor, en caso de ser posible.
 - Aplica para parámetros y valores de retorno

Tipos de datos

Las validaciones de los tipos se pueden dar de las siguientes dos maneras:

- Coercitiva (por omisión)
 - Hace castings implícitos. Se forza el valor, en caso de ser posible.
 - Aplica para parámetros y valores de retorno
- Estricta

Tipos de datos

Las validaciones de los tipos se pueden dar de las siguientes dos maneras:

- Coercitiva (por omisión)

Tipos de datos

Las validaciones de los tipos se pueden dar de las siguientes dos maneras:

- Coercitiva (por omisión)
- Estricta
 - Debe ser especificada una cláusula de declare dentro del archivo.
 - Afecta tanto a los parámetros como a los valores de retorno.
 - Espera un tipo igual al de la declaración.
 - El efecto estricto de las funciones es dado por el archivo donde fueron declaradas.

Y sólo aplica para los parámetros y valores de retorno de las funciones/métodos

Declaración por omisión

En los parametros de las funciones se puede escribir un valor por omisión

```
function foo(string $cadena = 'cadena') {  
    ...  
}
```

Números de parámetros variables

Las funciones pueden recibir un número de parámetros variables. Todos ellos se almacenan en un arreglo. Se utiliza el token (partícula del lenguaje).

```
function foo(...$cadena) {  
    ...  
}
```

Índice

- 1 Primer acercamiento
 - Aplicaciones Web
 - ¿Qué es PHP?

- 2 PHP
 - Tipos de datos
 - Estructuras de control de flujo
 - funciones

Estructuras de control de flujo

- ¿Qué se entiende por flujo?

Estructuras de control de flujo

- ¿Qué se entiende por flujo?
- ¿Que estructuras de control conocen?

Estructuras de control de flujo

El flujo se puede definir como la secuencia de instrucciones que se ejecutan.

Veremos:

- Condicionales
 - if then else
 - switch case

Estructuras de control de flujo

El flujo se puede definir como la secuencia de instrucciones que se ejecutan.

Veremos:

- Condicionales
 - if then else
 - switch case
- ciclos for
 - for each
 - while

Condicionales

if, else if ó elseif

Sintaxis:

```
if(cond1){
```

```
...
```

```
} else if(cond2){
```

```
...
```

```
} else{
```

```
...
```

```
}
```

Sintaxis:

```
if(cond1){
```

```
...
```

```
} elseif(cond2){
```

```
...
```

```
} else{
```

```
...
```

```
}
```

Condicionales

if,else if ó elseif

Sintaxis:

```
if(cond1):
```

```
...
```

```
elseif(cond2):
```

```
...
```

```
else:
```

```
...
```

```
endif;
```

Condicionales

switch case

Sintaxis:

```
switch(expresionSimple){  
    case Caso1:  
        break;  
    case Caso2:  
        break;  
    default:  
        break;  
}
```

Sintaxis:

```
switch(expresionSimple):  
    case Caso1:  
        break;  
    case Caso2:  
        break;  
    default:  
        break;  
endswitch;
```

Condicionales

expresionSimple es un tipo cadena, entero o flotante.

Ciclos

for

Sintaxis:

```
for(exp1;exp2;exp3){  
  
}
```

Sintaxis:

```
for(exp1;exp2;exp3):  
  
endfor;
```


Ciclos

exp1: valor inicial de iterador.

exp2: condición que se tiene que cumplir para cada iteración.

exp3: incremento del iterador.

Ciclos

while

Sintaxis:

```
while(cond){  
  
}
```

Sintaxis:

```
while(cond):  
  
endwhile;
```

Ciclos

foreach

Sintaxis:

```
foreach(array as $val){
```

```
}
```

Para arreglos asociativos

```
foreach(array as $key => $val){
```

```
}
```

También aplica la versión de los dos puntos.

Ciclos

foreach

Si los valores del arreglo fueron modificados dentro del ciclo, al terminar el ciclo no se verán reflejados esos cambios.

Para modificar el arreglo dentro del ciclo hay que usar referencias:

```
foreach(array as &$val){...}  
unset($val); //borramos la referencia.
```

Índice

1 Primer acercamiento

- Aplicaciones Web
- ¿Qué es PHP?

2 PHP

- Tipos de datos
- Estructuras de control de flujo
- **funciones**

funciones: Manejo de Excepciones

- **¿Qué es una excepción?**

Una excepción es la combinación de valores de entrada que producen un comportamiento anómalo en la función

funciones: Manejo de Excepciones

- **¿Qué es una excepción?**

Una excepción es la combinación de valores de entrada que producen un comportamiento anómalo en la función

- **¿A qué se refiere el evento de lanzar una excepción?**

Una función avisa que, bajo ciertos valores de entrada, tendrá un comportamiento anómalo

funciones: Manejo de Excepciones

- **¿Qué es una excepción?**

Una excepción es la combinación de valores de entrada que producen un comportamiento anómalo en la función

- **¿A qué se refiere el evento de lanzar una excepción?**

Una función avisa que, bajo ciertos valores de entrada, tendrá un comportamiento anómalo

- **¿Quién tiene que cachar una excepción**

Cuando un segmento de código hace uso de una función que lanzará una excepción.