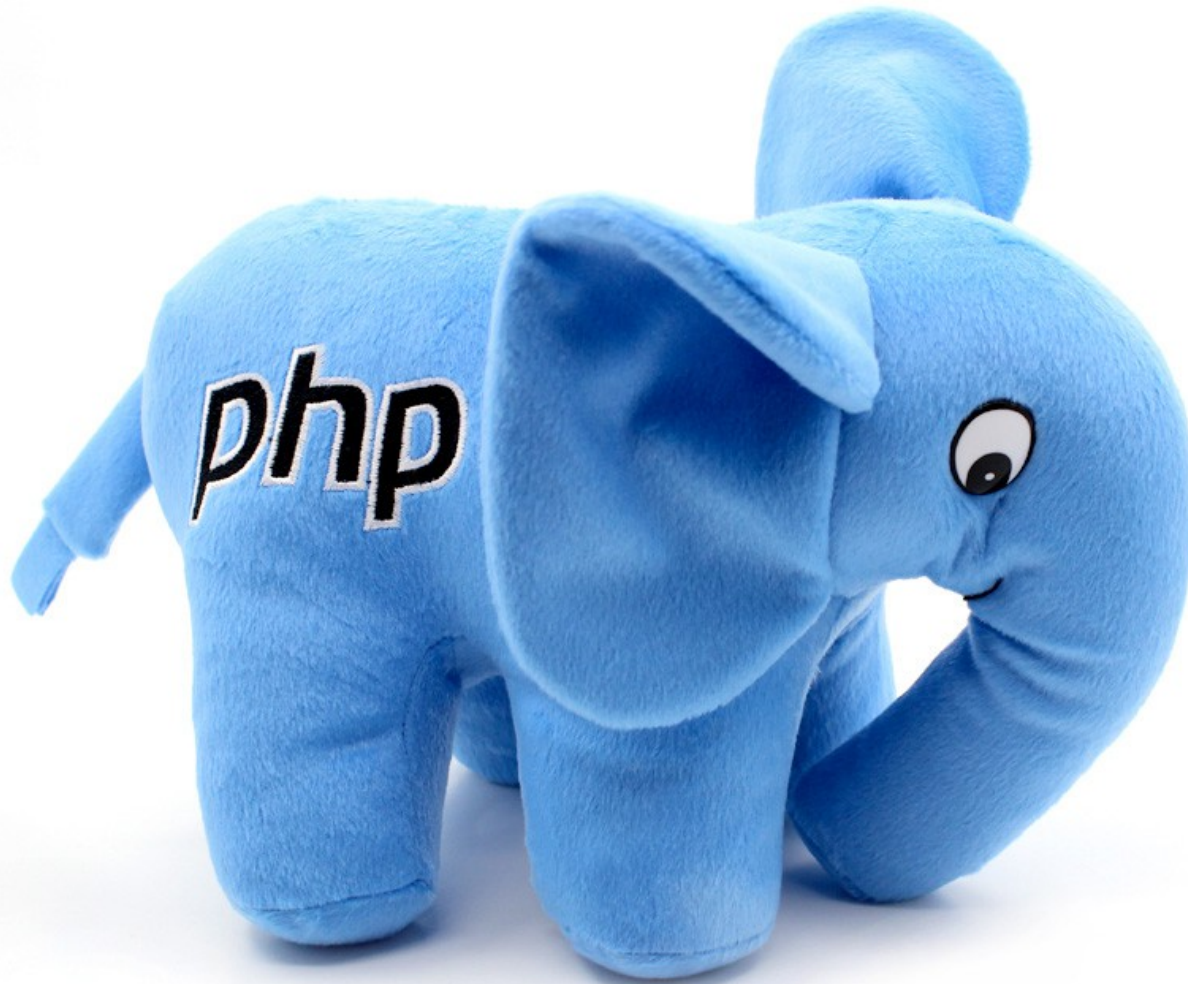


# Introducción a PHP

✱UTN



# Pre-Hypertext Processor (Procesador Previo al Hipertexto)

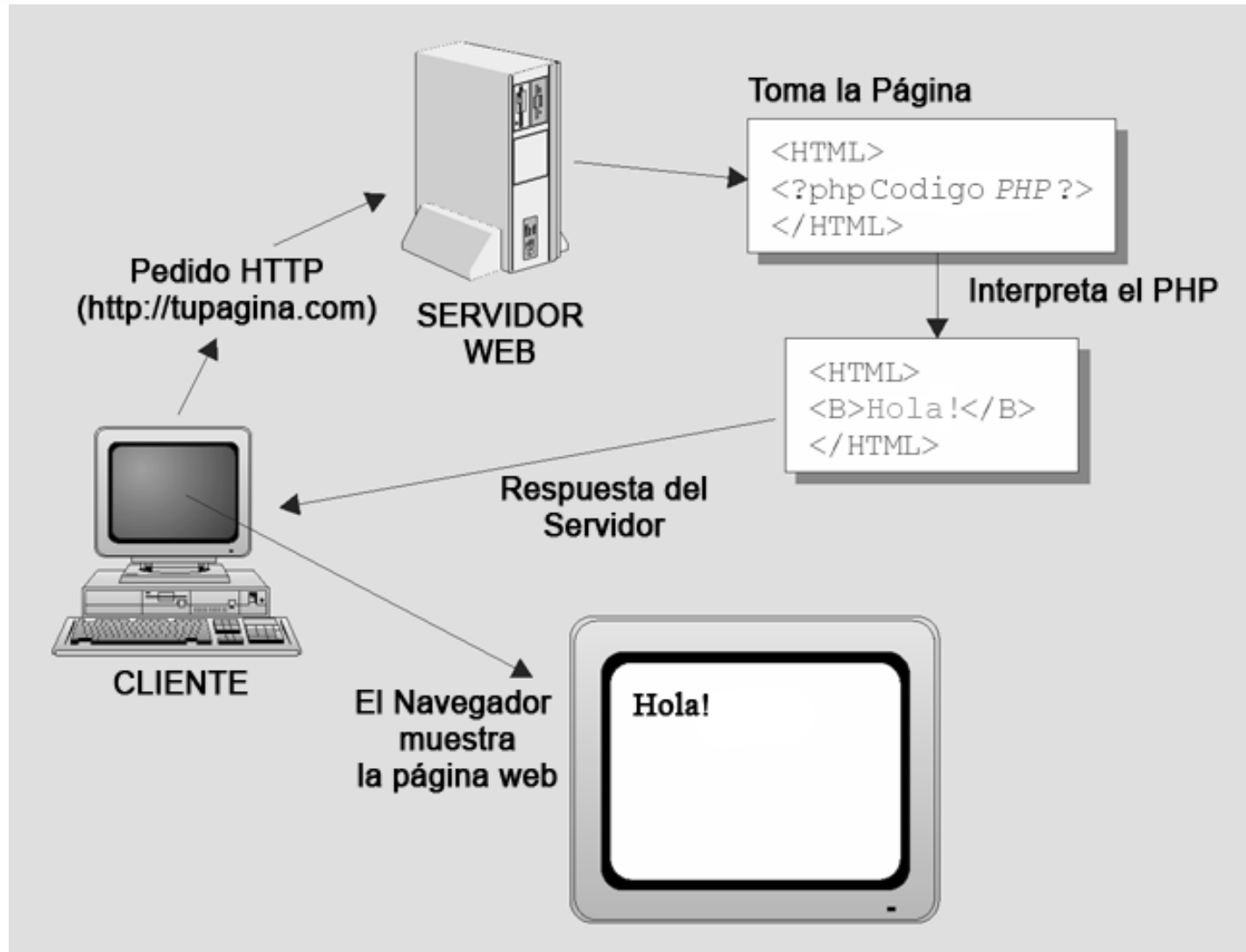


- Es un lenguaje de programación utilizado para el diseño de sitios web dinámicos, con la posibilidad de combinarlo con bases de datos.
- Los script de PHP se intercalan en el código HTML,
  - el encargado de procesar esos scripts es el intérprete del servidor web.



- PHP es un **lenguaje ejecutado en el servidor**
  - lo que se codifique en PHP, no va a ser interpretado por el Navegador, si no por un Servidor Web que procese esas instrucciones y devuelva al cliente el código HTML resultante.





- Logra un mayor dinamismo del contenido
- Combinándolo con la gestión de bases de datos, lograr hacer funcionar algunos de los siguientes ejemplos:
  - Foros
  - Guestbooks
  - Blogs
  - Calendarios
  - Sistemas de Carrito de Compras
  - Restricción de acceso a sitios web mediante user/password
  - Registro de estadísticas
  - Proceso de Formularios de Email



- Nuestros archivos deben llevar la **extensión .php** para que el intérprete los procese.



- Debemos acceder a estos archivos PHP a través de un servidor web
- Se pueden intercalar órdenes en lenguaje PHP alternándolas dentro de una página escrita en lenguaje HTML, tantas veces como sea necesario



**<?php  
// código  
?>**

**Funciona  
siempre.**

**<? ?>** No es estándar, aunque sea utilizado de esta manera.  
No todas las configuraciones de **php** habilitan su uso.

**<script language="php">  
</script>**

**<% %>** Estilo ASP, no es estándar, depende de la configuración del intérprete.





- Las sentencias siempre finalizan con un punto y coma;
- El código PHP siempre debe estar encerrado por sus delimitadores ( `<?php` y `?>` )
- Los **comentarios de una sola línea** en PHP pueden hacerse empezando la línea con `//` o con `#`
- Los **comentarios de múltiples líneas** están encerrados entre `/*` y `*/`



Para agregar un código PHP dentro de una página HTML debemos, al crear el archivo, definirlo con extensión php y dentro del contenido de la página, encerrar el programa entre los símbolos **<?php** [script PHP] **?>**.

```
<html>
    <head>
</head>
    <body>
        <?php
            echo "Hola Mundo";
        ?>
    </body>
</html>
```



- Una variable en PHP es un espacio de memoria que guarda un valor, y tiene un identificador (nombre de la variable) y un tipo de dato.
- No es necesario indicar de qué tipo de dato es una variable, el intérprete de PHP se dara cuenta según su contenido.
- Se utiliza para guardar información en ellas que luego usaremos varias veces a lo largo del código de la página.



- **SIEMPRE** llevan el signo “\$” adelante.
- Nombres solo con: **letras, números y \_** (guión bajo)
- **NUNCA** pueden llevar espacios ni empezar con un número, pero **SÍ** pueden empezar con letras o con \_
- Los nombres de las variables **son sensibles a mayúsculas y minúsculas** (case-sensitive).



# Variables – usos válidos e inválidos



## Nombres válidos

**\$variable**  
**\$var1**  
**\$var\_1**  
**\$VAR**  
**\$vAR**  
**\$\_variable**

## Nombres no válidos

**\$1**  
**\$666**  
**\$-variable**  
**\$2variable**  
**\$ variable**  
**\$'variable**  
**\$>variable**  
**\$<variable**



*Las variables se declaran cuando se le asigna un valor, por ejemplo:*

**\$dia = 24;** //variable de tipo numerica.

**\$sueldo = 758.43;** //variable de tipo double.

**\$nombre = "juan";** //variable de tipo string (cadena).

**\$salida = true;** //variable de tipo boolean.



## Ejemplos:

```
<?php
$Cadena = "Hola Mundo"; //Esto es
    correcto
$cadena = "Hola Mundo"; /*$Cadena y
    $cadena son dos variables distintas */
$_Cadena = "Hola Mundo"; //Esto es
    correcto
$1Cadena = "Hola Mundo"; //Esto NO es
    correcto
$Una Cadena = "Hola Mundo"; //Esto
    TAMPOCO es correcto.
?>
```



- **Operadores aritméticos**

- + Suma dos valores
- Resta dos valores (o pasa a negativo un valor)
- \* Multiplica dos valores
- / Divide dos valores
- % Resto de dividir dos valores





- **Operadores de comparación**

**==** Comprueba si dos números son iguales

**!=** Comprueba si dos números son distintos

**>** Mayor que, devuelve true en caso afirmativo

**<** Menor que, devuelve true en caso afirmativo

**>=** Mayor o igual

**<=** Menor o igual



- **Operadores lógicos**

**!** Operador NO o negacion. Si era true pasa a false y viceversa

**And** Operador Y, si ambos son verdaderos vale verdadero

**or** Operador O, vale verdadero si alguno de los dos es verdadero

**xor** Verdadero si alguno de los dos es true pero nunca ambos

**&&** True si ambos lo son

**||** True si alguno lo es



- Operadores de incremento

**++ \$variable**

Aumenta de 1 el valor de \$variable

**-- \$variable**

Reduce de uno el valor de \$variable



- **Operadores lógicos**

**=** Asigna a la parte derecha el valor izquierdo

**+=** Realiza la suma de la derecha con la izquierda y la asigna a la derecha

**-=** Realiza la resta de la derecha con la izquierda y la asigna a la derecha

**\*=** Realiza la multiplicación de la derecha con la izquierda y la asigna a la derecha

**/=** Realiza la división de la derecha con la izquierda y la asigna a la derecha

**%=** Se obtiene el resto y se asigna



Cuando se pretende que el script tome un camino concreto en determinados casos y otro diferente si las condiciones de ejecución difieren, se utiliza el conjunto de instrucciones: **if**, **else** y **elseif**.

La estructura base es la siguiente:

```
if (Condición) {  
    Instrucción 1;  
    Instrucción 2;  
}  
else {  
    Instrucción a;  
    Instrucción b;  
}
```



Cuando se pretende que el script tome un camino concreto en determinados casos y otro diferente si las condiciones de ejecución difieren, se utiliza el conjunto de instrucciones: **if**, **else** y **elseif**.

La estructura base es la siguiente:

```
if (Condición) {  
    Instrucción 1;  
}  
else {  
    Instrucción 2;  
}
```

Si la condición es verdadera se ejecutará la instrucción 1, si es falsa se ejecutará la instrucción 2.



Otra forma de sintaxis:

```
$variable = condicion ? Valor si es verdadera : valor si es falsa;
```

Ejemplo:

```
<?php  
$tmp = 1;  
$variable = $tmp>5 ? 'Es mayor a 5' : No es mayor a 5';  
echo $variable;  
?>
```



# Condicionales elseif

```
if (Condicion 1)
{
    Instrucción a;
}
elseif (Condicion 2)
{
    Instrucción b;
}
else
{
    Instrucción c;
}
```

Si la Condición1 no se cumple, pasa a la Condición2, si es verdadera se ejecutará la instrucción b, si es falsa se ejecutará la instrucción c.

Si se cumple la Condición 1, se ejecutará la Instrucción a.

**Recordar los operadores:**

**==** igual.

**!=** distinto.

**>=** mayor o igual.

**>** mayor.

**<=** menor o igual

**<** menor





- Un switch busca dentro de los “case” el valor de la variable o expresión evaluada (generalmente se evalúan variables).
- Si lo encuentra, ejecuta el código correspondiente.
- Si no lo encuentra, ejecuta el código por default.
- *Observaciones:* El case por default es opcional.
- Es importante poner “break;” al final de cada bloque de código dentro de cada case para que el switch no siga comparando al valor de la variable con los case que le siguen al correcto.



- Un switch busca dentro de los “case” el valor de la variable o expresión evaluada (generalmente se evalúan variables).
- Si lo encuentra, ejecuta el código correspondiente.
- Si no lo encuentra, ejecuta el código por default.
- *Observaciones:* El case por default es opcional.



Es importante poner “break;” al final de cada bloque de código dentro de cada case para que el switch no siga comparando al valor de la variable con los case que le siguen al correcto.

```
switch(expresión) {  
    case "a": //Código a ejecutar;  
                break;  
    case "b": //Código a ejecutar;  
                break;  
    case "c": //Código a ejecutar;  
                break;  
    default: //Código a ejecutar por  
default.  
}
```

