

CONCEPTOS BÁSICOS PHP

Sintaxis y Variables

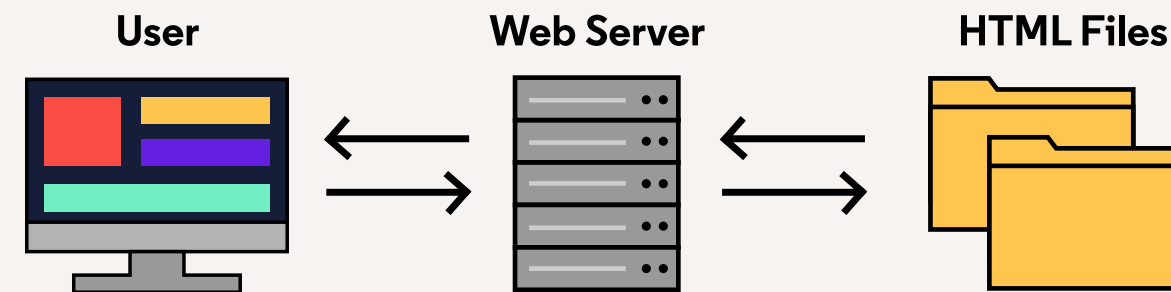
Producción Multimedia



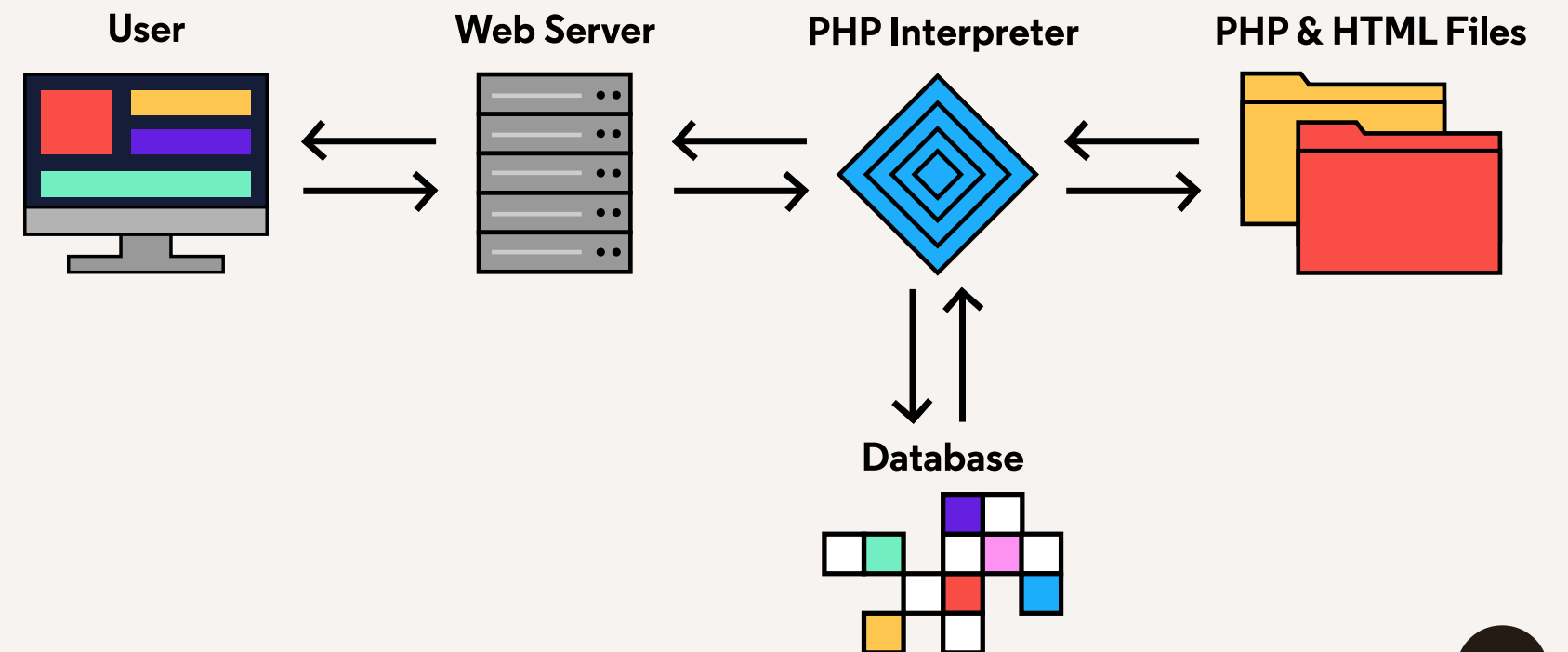
PHP y HTML

Con PHP podremos crear páginas web dinámicas. Estas son aquellas en las que cada visitante del sitio web obtiene una página personalizada que puede tener una apariencia diferente a la de otro visitante. Esto contrasta con las páginas web estáticas que proporcionan el mismo contenido a cada visitante.

Static



Dynamic



¿Qué se necesita para empezar?

1. archivo.php

- No requiere licencia
- Editor de texto: Visual Studio Code

2. Servidor

- Servidor Local: XAMPP, MAMP, WAMP

3. Base de Datos

- MySQL



Sintaxis

PHP fue diseñado para trabajar en integración con HTML. Cuando el sitio web se entrega desde el back-end al front-end, el contenido PHP se ejecuta y se agrega al HTML .

- Para empezar a escribir PHP se deben establecer las etiqueta de apertura **<? php** y el cierre con **?>**.
- Generalmente, PHP ignora los espacios en blanco (tabulaciones, espacios, nuevas líneas).
- A diferencia de muchos otros lenguajes, PHP no siempre distingue entre mayúsculas y minúsculas, por lo que **Echo** es una declaración válida. Sin embargo, es una buena práctica escribir en minúsculas.
- Se requiere un punto y coma (;) al final de cada instrucción en PHP.
- Se puede incluir PHP en cualquier parte del archivo HTML.

```
<?php  
echo "Hello, World!";  
?>
```



Mostrar contenido con "echo"

En PHP, la palabra reservada **echo** se usa para generar texto. En este caso, el texto es todo lo que está entre comillas dobles (").

```
<p>Este párrafo se interpreta como HTML</p>
<?php echo "<p>Pero este texto es interpretado por PHP y
entregado como HTML</p>";?>
<?php echo "<ul><li>Se puede utilizar cualquier etiqueta
HTML, </li> <li> como esta lista. </li> </ul>";?>
<?php echo "<h2> Esto será un subtítulo en HTML</h2>" ?>
```



Comentarios

Comentarios de una sola línea:

- Por lo general, se utilizan para explicaciones breves o puntos de aclaración.
- Se puede usar `#` o `//` para crear un comentario de una sola línea.
- Cualquier cosa en la misma línea después de estos símbolos no es ejecutada por PHP.

```
// Comentario de una sola línea  
echo "Hola Mundo!"; # Imprimir código
```

```
/* "I've never thought of PHP as more  
than a simple tool to solve problems."  
- Rasmus Lerdorf */  
echo "\nRasmus is the creator of PHP!";
```

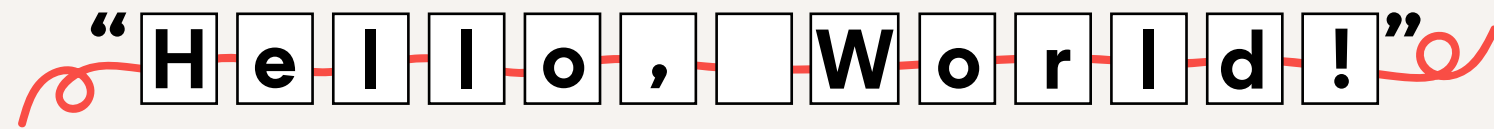
Comentario de varias líneas:

- Esto se usa para descripciones más largas, una guía más detallada sobre cómo usar correctamente la sección de código.
- Estos comentarios comienzan con `/*` y terminan con `*/`.



Strings

- Son secuencias de caracteres.
- Puede tener cualquier longitud y contener letras, números, símbolos o espacios entre comillas.
- Se usan comillas dobles o simples para indicar el principio y final de un string.



“H e l l o , W o r l d !”

- Pueden incluir secuencias de escape como `\` o `\n`

En el siguiente ejemplo se produciría un error: **"Ella le dijo "hola" al perro".**

Se puede corregir usando secuencias de escape o alternando comillas dobles y simples:

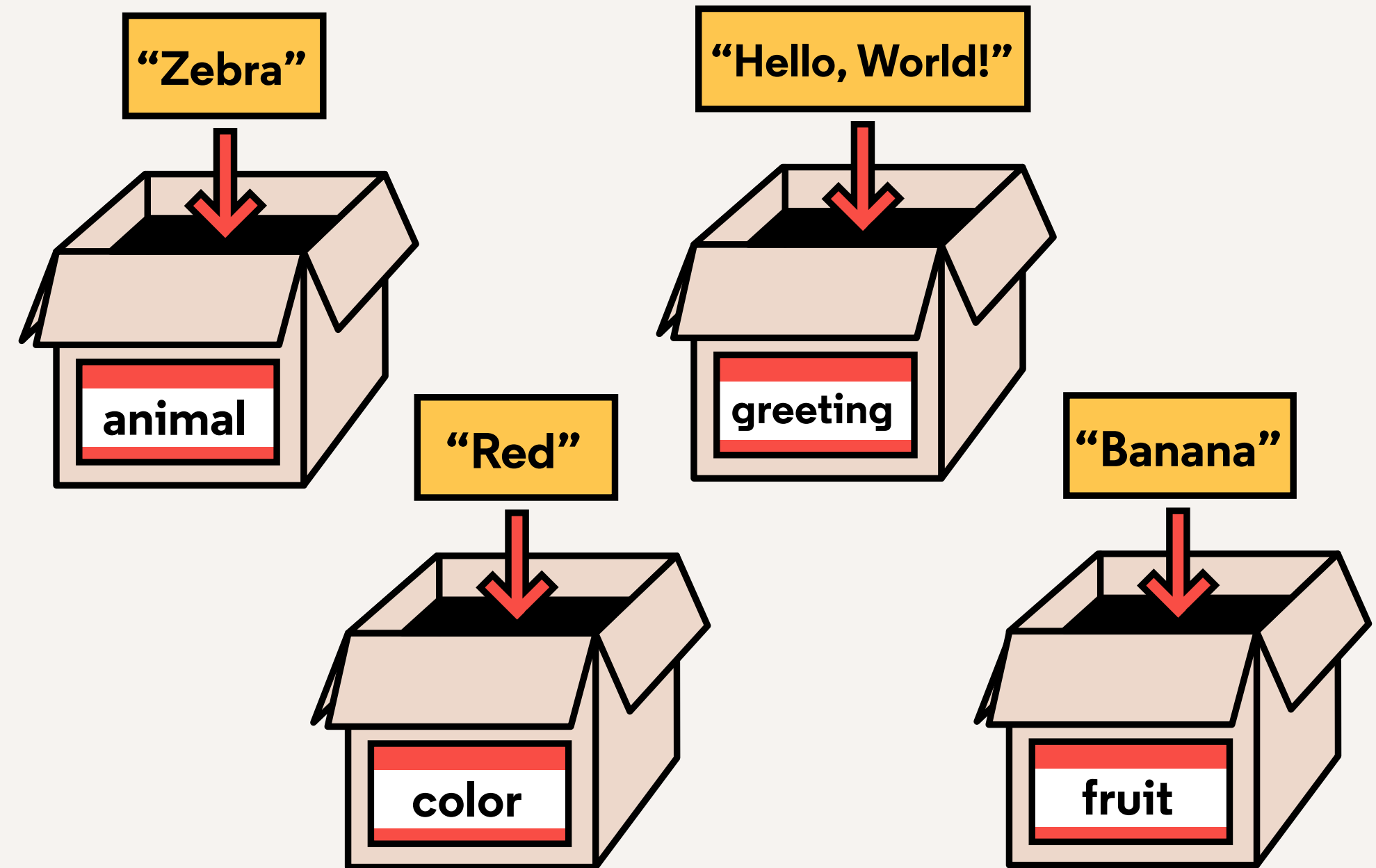
'Ella le dijo "hola" al perro'; o **"Ella le dijo \"hola\" al perro"**



Variables

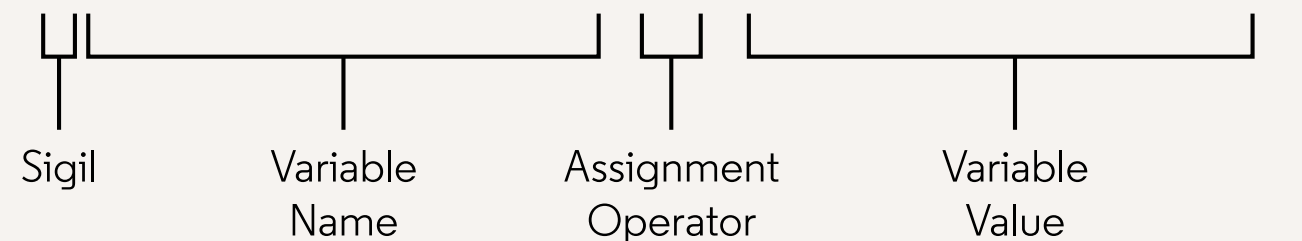
Con las variables, almacenamos valores para que podamos reutilizarlos fácilmente en un programa.

Antes de que podamos usar variables en nuestro código, debemos declararlas y asignarlas.



Reglas Sintaxis Variables

`$my_example = "my value";`



Sigil Variable Name Assignment Operator Variable Value

- Para declarar variables se debe empezar con el signo peso (**\$**) seguido del nombre de la variable.
- Para asignarle un valor usamos el operador de asignación (**=**) seguido del valor que estamos asignando a la variable.
- Los nombres de las variables pueden contener números, letras y guiones bajos (**_**),
- Deben comenzar con una letra o un guión bajo.
- Los nombres de las variables distinguen entre mayúsculas y minúsculas, lo que significa que PHP tratará las variables `$ my_example` y `$ My_example` como dos variables diferentes.

Concatenando Strings y Variables

Por medio del operador de concatenación (.) podremos combinar dos o más cadenas de texto o variables.

Ejemplo 1:

```
echo "one" . " " . "two" . " " . "three"; //
```

Prints: one two three

Ejemplo 2:

```
$dog_name = "Tadpole";  
echo "My dog is named " . $dog_name;  
// Prints: My dog is named Tadpole
```



- Las cadenas de PHP nos permiten colocar variables directamente en cadenas entre comillas dobles.
- Estas variables se analizarán, lo que significa que se leerá como el valor que tienen en lugar de verlas solo como una secuencia de caracteres.

Ejemplo:

```
$dog_name = "Tadpole";  
$favorite_food = "salmon";  
$color = "brown";  
  
echo "I have a $color dog named $dog_name and her  
favorite food is $favorite_food."  
// Prints: I have a brown dog named Tadpole and her  
favorite food is salmon.
```



Ejercicio

Cree una página web y llámela **saludo.php**

Defina las variables **\$saludo**, **\$nombre1** y **\$nombre2**

Imprima las tres variables juntas para armar un saludo completo que diga:

Hola "Pedro" y "Ana", ¿Cómo están?

Al final imprima una línea de división horizontal con HTML y debajo imprima el siguiente texto:

Página creada por:

– **"Pedro"**

– **"Ana"**



Reasignación de Variables

El proceso de asignar un nuevo valor a una variable se llama reasignación.

Reasignamos una variable usando el operador de asignación en una variable que ya ha sido declarada:

```
$comida_favorita = "Tacos";  
echo $comida_favorita; // Prints: Tacos  
  
// Reasignar el valor de $comida_favorita a un nuevo string  
$comida_favorita = "Pizza";  
echo $comida_favorita; // Prints: Pizza
```

A menudo es útil crear nuevas variables asignadas al mismo valor que una variable existente:

```
$first_player_rank = "Beginner";  
$second_player_rank = $first_player_rank;
```



Operador de asignación y concatenación

El operador de asignación y concatenación (.=) es una forma reducida de concatenar una cadena de caracteres a una variable que ya tiene un valor asignado.

```
$parrafo = "Un elefante se balanceaba ";  
echo $parrafo;  
  
$parrafo .= "sobre la tela de una araña";  
echo $parrafo;
```



Operadores Aritméticos

- $+$ Suma
- $-$ Resta
- $*$ Multiplicación
- $/$ División

Operadores Aritméticos de Asignación

- $=$ Asignación
- $+=$ Suma y Asignación
- $-=$ Resta y Asignación
- $*=$ Multiplicación y Asignación
- $/=$ División y Asignación



Orden de Preferencia

```
$resultado;
```

```
$resultado = 5 + 5 * 10;
```

```
// 50 primero * y /
```

```
// 55 después + y -
```

```
$resultado = ( 5 + 5 ) * 10;
```

```
// Resultado: 100
```


Operadores de incremento / decremento

<code>++\$x</code>	Pre-incremento	Incrementa \$ x en uno, luego devuelve \$ x
<code>\$x++</code>	Post-incremento	Devuelve \$ x, luego incrementa \$ x en uno
<code>--\$x</code>	Pre-decremento	Disminuye \$ x en uno, luego devuelve \$ x
<code>\$x--</code>	Post-decremento	Devuelve \$ x, luego decrementa \$ x en uno



Ejercicio: Truco de "magia" matemática:

- Cree una variable, **\$mi_numero** y asigne como valor cualquier número.
- A continuación, cree una segunda variable, **\$respuesta** y asigne **\$mi_numero** como su valor.
- Utilice el operador de suma y asignación para agregar 2 a **\$respuesta**
- Utilice el operador de multiplicación y asignación para multiplicar **\$respuesta** por 2.
- Utilice el operador de resta y asignación para restar 2 de **\$respuesta**.
- Utilice el operador de división y asignación para dividir **\$respuesta** por 2.
- Utilice el operador de resta y asignación para restar su número original (**\$mi_numero**) de **\$respuesta**.
- Finalmente, use echo para imprimir **\$respuesta** en el navegador.

Respuesta debe ser 1



Tipos de Datos

String

Es una secuencia de caracteres, como "¡Hola mundo!".

```
$x = "Hello world!";
```

Integer

Valor numérico entero. Puede ser positivo o negativo

```
$x = 5985;
```

Float

Valor numérico decimal. Puede ser positivo o negativo

```
$x = 10.365;
```

Boolean

Un booleano representa dos estados: **TRUE** o **FALSE**.

```
$x = true; $y = false;
```

Array

Almacena varios valores en una sola variable.

```
$cars = array("Volvo", "BMW", "Toyota");
```

Null

Es una variable que no tiene ningún valor asignado.

```
$x = null;
```



Constantes

Las constantes son como variables, excepto que una vez definidas no se pueden cambiar. A diferencia de las variables, las constantes son automáticamente globales en todo el script.

Sintaxis

```
define ( name, value, case-insensitive );
```

name: especifica el nombre de la constante. Sin signo **\$** antes del nombre constante

value: especifica el valor de la constante

case-insensitive: especifica si el nombre de la constante no debe distinguir entre mayúsculas y minúsculas. El valor predeterminado es falso



Ejemplo Constantes

Ejemplo 1:

```
define("SALUDO", "Bienvenido a PHP");  
echo SALUDO;
```

Ejemplo 2:

```
define("SALUDO", "Bienvenido a PHP", true);  
echo saludo;
```

