**ANTONIO VALERO MALDONADO**

**1. ¿Quién inventó PHP?**

PHP fue creado en **1994 por Rasmus Lerdorf**, un programador danés-canadiense.

**2. ¿Qué es PHP?**

PHP es un **lenguaje de programación de código abierto** especialmente diseñado para el **desarrollo web**. Se ejecuta en el **lado del servidor** y permite crear páginas dinámicas e interactivas.

**3. ¿Qué significa PHP?**

Originalmente significaba **“Personal Home Page”**, pero hoy en día significa **“PHP: Hypertext Preprocessor”** (nombre recursivo).

**4. ¿Cuáles son las diferencias más notables entre JavaScript y PHP?**

| **Característica** | **PHP (Servidor)** | **JavaScript (Cliente)** |
| --- | --- | --- |
| **Ejecución** | Lado del servidor | Lado del cliente (navegador) |
| **Tipo** | Lenguaje interpretado para backend | Lenguaje interpretado para frontend |
| **Instalación** | Requiere servidor (Apache, Nginx, etc.) | No requiere servidor, se ejecuta en navegador |
| **Uso principal** | Generar contenido dinámico en el servidor | Interactividad y manipulación del DOM en la página |
| **Bases de datos** | Integración nativa (MySQL, PostgreSQL) | Necesita APIs/servicios para conectarse |

**5. ¿Qué tipo de licencia tiene PHP?**

PHP está bajo la **Licencia PHP**, una licencia **libre y de código abierto**, compatible con la **GNU GPL** en algunos aspectos.

**6. ¿Dónde podemos encontrar ayuda para PHP?**

* **Sitio oficial:** <https://www.php.net/manual/es/>
* Foros y comunidades: Stack Overflow, Reddit, PHP Freaks, etc.
* Documentación de frameworks (Laravel, Symfony, etc.).

**7. ¿Qué necesitamos para trabajar con PHP?**

* Un **servidor web** (como Apache o Nginx).
* **PHP instalado** en el servidor.
* Opcional: un **gestor de bases de datos** (MySQL, MariaDB, etc.).
* Un editor de código (VS Code, Sublime Text, etc.).
* Paquetes como **XAMPP, WAMP o Laragon** facilitan la instalación de todo esto.

**8. ¿Cómo accedemos a nuestro servidor local?**

En tu navegador, escribes:

* http://localhost
* o http://127.0.0.1

Ahí verás la página principal del servidor local (por ejemplo, de XAMPP).

**9. ¿Dónde se encuentra la página de inicio de XAMPP?**

Normalmente en:

* http://localhost/xampp/  
  Ahí está el panel de administración de XAMPP (Apache, MySQL, PHP, etc.).

**PRACTICA 2:**

**¿Qué es lo que hace la función phpinfo()?**

La función **phpinfo()** en PHP es una **función interna** que muestra en pantalla **toda la información de configuración del entorno PHP** que está ejecutándose en tu servidor.

Te dice **la versión de PHP instalada**.

Lista **las extensiones cargadas** (MySQLi, cURL, GD, etc.).

Muestra **las variables de entorno** y **configuraciones del php.ini**.

Informa sobre el **servidor web** (Apache, Nginx, etc.).

Enseña **los valores de las directivas** de PHP (límite de memoria, tamaño máximo de subida.

**¿Cómo comenzamos un bloque de PHP?**

El bloque de código PHP se abre y se cierra con **etiquetas especiales**:

<?php

?>

**¿Cómo imprimimos un mensaje en pantalla?**

En PHP puedes usar principalmente dos instrucciones:

<?php

echo "Hola mundo"; // Imprime texto en pantalla

print "Hola mundo"; // También imprime texto

?>

echo es la más común (puede imprimir varias cadenas separadas por comas).

print funciona casi igual, pero devuelve un valor (1), por eso es más lenta.

**¿Cuál es la diferencia con el ejercicio 2?**

Introdujo una etiqueta html en el echo.

**¿Cómo declaramos una variable en PHP?**

* En PHP **todas las variables comienzan con el signo $** seguido del nombre.
* No es necesario especificar el tipo (PHP es de **tipado dinámico**).

¿**Importan las mayúsculas y minúsculas en la declaración de las variables?**

* **Sí, en PHP las variables son sensibles a mayúsculas y minúsculas** (case-sensitive).

**¿Qué es la concatenación?**

La **concatenación** es **unir cadenas de texto** (strings) y/o variables para formar una sola cadena.  
Por ejemplo: unir "Hola " + "Mundo" = "Hola Mundo".

Es muy común cuando quieres mostrar mensajes que incluyan variables.

**¿Cómo hacemos la concatenación en PHP?**

En PHP se hace con el **operador punto .**

Puedes concatenar tantas cadenas y variables como quieras

**Qué hace este script:**

1. **Crea una página HTML** con un título <h1> que dice *“Break en la ventana del navegador”*.
2. **Dentro del párrafo <p> ejecuta código PHP**:
   * Declara la variable $Name con el valor "Miguel".
   * Usa echo para imprimir:
     + "Hola <b>Miguel</b>, encantado de conocerte"
     + Un salto de línea en HTML con <br> (eso es el “break” que ves en el navegador).
     + Después imprime "Gracias por venir!" en la línea siguiente.

**muestra un saludo personalizado usando PHP** y hace un **salto de línea** con <br> para que el segundo mensaje salga debajo del primero.

En el ejercicio 6 solo usamos <br> para hacer un salto de línea visible en el navegador.  
En este ejercicio se utiliza <br>\n y \n, que además de mostrar el salto visible añaden un salto de línea en el **código fuente HTML**.  
El resultado en pantalla es casi igual, pero el HTML generado queda más ordenado.

**¿Cómo se declara una variable numérica en PHP?**

Solo asignas un número a una variable con $ (PHP detecta el tipo automáticamente):

**¿Cómo se realiza una operación con variables numéricas?**

Puedes sumar, restar, multiplicar, dividir usando los operadores aritméticos: +, -, \*, /.

**¿Es posible mostrar en pantalla en una misma línea cadenas y números?**

Sí, usando concatenación con . o interpolación dentro de comillas dobles.  
Ejemplo: echo $nombre . " tiene " . $edad . " años"; o echo "$nombre tiene $edad años";

**¿Cómo incrementamos una variable?**

Con el operador ++ o sumando directamente:

EJ 11

**==**  
• **Respuesta:** Igual a (compara valores, no tipos).

**!=**  
• **Respuesta:** Distinto de (diferente valor).

**<**  
• **Respuesta:** Menor que.

**>**  
• **Respuesta:** Mayor que.

**>=**  
• **Respuesta:** Mayor o igual que.

**<=**  
• **Respuesta:** Menor o igual que.

**¿A qué equivale &&?**

**&&**  
• **Respuesta:** Y lógico (AND). Devuelve verdadero solo si **ambas condiciones** son verdaderas.

**¿A qué equivale ||?**   
**||**  
• **Respuesta:** O lógico (OR). Devuelve verdadero si **al menos una** de las condiciones es verdadera.

**1. ¿Qué funciones de PHP usamos en el script?**

En el script se usan **las siguientes funciones de PHP**:

1. **round()** → Redondea un número a una cantidad específica de decimales.
   * Ejemplo: round($resultado, 2) redondea $resultado a 2 decimales.
2. **sprintf()** → Formatea un número o cadena según un formato especificado, devolviendo el resultado como cadena.
   * Ejemplo: sprintf("%01.2f", $resultado) muestra $resultado con 2 decimales y relleno mínimo de 1 dígito.
3. **echo** → Imprime texto, números o resultados en la página web.

**Qué hace la sentencia IF...ELSE?**

La sentencia **if...else** en PHP sirve para **tomar decisiones según una condición**:

* **if**: Evalúa una condición; si es verdadera, ejecuta el bloque de código dentro de if.
* **else**: Se ejecuta si la condición del if es falsa.

En el ejemplo:

* Se compara $a < $b.
* Si $a es menor que $b, muestra "a es menor que b".
* Si no, muestra "a no es menor que b".

La diferencia principal entre **switch** y **if** en PHP es la forma en que evalúan condiciones:

* **if**: Evalúa **expresiones booleanas complejas** y permite múltiples comparaciones con operadores lógicos (&&, ||, >, <, etc.).
* **switch**: Evalúa **el valor de una variable o expresión** y lo compara **exactamente** con distintos casos (case). Es más limpio cuando se tienen muchas comparaciones directas de un mismo valor.

En el ejemplo:

* switch($posicion) revisa si $posicion es "arriba", "abajo" o cualquier otro valor (default).
* Es equivalente a varios if...elseif...else, pero más **ordenado y legible** cuando hay muchos posibles valores fijos.