

Variables y Tipos de Datos

Crea un programa que defina tres variables (nombre, edad y ciudad) y muestre su contenido en pantalla.

Declara una variable con un número decimal y conviértela a entero usando *casting*.

Crea un script que sume dos números y muestre el resultado con un texto explicativo.

Declara una variable de tipo booleano y usa `echo` para mostrar “Verdadero” o “Falso” según su valor.

Define una cadena de texto y muestra su longitud, su versión en mayúsculas y minúsculas.

Crea una frase con variables interpoladas (usando comillas dobles) que muestre nombre y edad de una persona.

Crea un array con tres nombres y muestra cada uno en una línea diferente usando `echo`.

Operadores

Crea un programa que determine si un número es par o impar utilizando el operador módulo `%`.

Declara dos números y muestra si el primero es mayor, menor o igual al segundo.

Usa operadores lógicos (`&&`, `||`) para comprobar si una persona tiene entre 18 y 65 años y puede trabajar.

Estructuras condicionales (IF / ELSE / ELSEIF / SWITCH)

Pide (o define) una nota y muestra si el alumno está suspenso, aprobado, notable o sobresaliente.

Define una variable con el nombre de un día de la semana y usa `switch` para mostrar un mensaje según el día.

Crea un programa que determine si una temperatura está por debajo de 0, entre 0 y 30 o por encima de 30 grados.

Simula un semáforo: según el color almacenado en una variable (“rojo”, “amarillo” o “verde”), muestra una acción.

Usa una condición `if` para comprobar si una contraseña introducida coincide con la guardada.

Bucles (WHILE, FOR, DO WHILE)

Muestra por pantalla los números del 1 al 10 usando un bucle `while`.

Calcula la suma de los números del 1 al 100 usando un bucle `for`.

Muestra todos los elementos de un array con un `while` o `foreach`.

Crea un bucle que genere una lista HTML (``) con los números del 1 al 5.

Usa un bucle `do while` para pedir un número hasta que el usuario introduzca el valor 0 (simula la entrada con variables).

Ejercicio 1: Tarjeta de Perfil Dinámica

Objetivo: Usar variables PHP para rellenar una tarjeta de perfil HTML y mostrar/ocultar información con un if.

Fichero: perfil.php

Crea una estructura HTML5 completa.

Dentro del <body>, crea un bloque PHP y define las siguientes variables:

\$nombre_usuario = "Tu Nombre";

\$profesion = "Estudiante DAW";

\$edad = 20; (pon tu edad)

\$tiene_github = true; (tipo booleano)

En la parte HTML (fuera del bloque PHP, pero dentro del <body>), crea la estructura de la tarjeta:

Un <h1> que muestre el \$nombre_usuario.

Un <h2> que muestre la \$profesion.

Un <p> que diga "Edad: X años", mostrando la \$edad.

Lógica Condicional: Justo debajo del párrafo de la edad, añade un nuevo bloque PHP que use una estructura if:

Si \$tiene_github es true, debe imprimir un enlace () que diga "Ver Perfil de GitHub".

Si es false, no debe imprimir nada.

Ejercicio 2: Calculadora de Precios de Tienda

Objetivo: Usar if/elseif/else y operadores lógicos (&&) para calcular el precio final de un producto.

Fichero: tienda.php

Crea una estructura HTML5 completa y ponle el título "Calculadora de Precios".

Dentro del <body>, crea un bloque PHP y define las siguientes variables:

\$precio_base = 70.00; (tipo float)

\$categoria = "ropa"; (tipo string)

\$es_cliente_vip = false; (tipo booleano)

Lógica de Descuentos: Debes calcular un descuento basándote en estas reglas:

Si la \$categoria es "electronica", el descuento es del 15%.

Si la \$categoria es "ropa" Y (&&) el \$precio_base es mayor de 50€, el descuento es del 10%.

Si \$es_cliente_vip es true, se aplica un 5% de descuento adicional (acumulable).

En cualquier otro caso, no hay descuento.

Calcula el \$precio_final y muestra un resumen por pantalla:

Imprime "Precio Base: \$precio_base €"

Imprime "Categoría: \$categoria"

Imprime "Cliente VIP: Si/No" (usa un if para imprimir "Si" o "No")

Imprime "Precio Final: \$precio_final €" en un <h3>.

Ejercicio 3: Generador de Calendario Mensual Básico

Objetivo: Usar un bucle while para generar HTML dinámicamente y un if dentro del bucle.

Fichero: calendario.php

Crea una estructura HTML5 completa. Añade el siguiente <style> en tu <head> para que se vea bien:

```
<style>

    body { font-family: sans-serif; }

    .calendario { display: flex; flex-wrap: wrap; width: 500px; border: 1px solid #ccc; }

    .dia { width: 60px; height: 60px; border: 1px solid #eee; padding: 5px; box-sizing: border-box; }

    .hoy { background-color: #ffc; font-weight: bold; }

</style>
```

En el <body>, crea un <h1> que diga "Calendario de Noviembre".

Crea un bloque PHP y define estas variables:

```
$dias_del_mes = 30;

$día_actual = 1; // Este será tu contador

$día_de_hoy = 11; // El día que quieres resaltar
```

Lógica de Bucle:

Imprime la etiqueta de apertura del calendario: echo '<div class="calendario">';

Crea un bucle while que se ejecute mientras \$día_actual sea menor o igual que \$dias_del_mes.

Dentro del bucle:

Usa un if para comprobar si \$día_actual es igual que \$día_de_hoy.

Si lo es, imprime un <div> con dos clases: echo '<div class="día hoy">';

Si no lo es (else), imprime un <div> normal: echo '<div class="día">';

Imprime el número del día (la variable \$día_actual).

Cierra el </div>: echo '</div>';

¡Importante! Incrementa tu contador: \$día_actual++;

Fuera del bucle:

Cierra el div del calendario: echo '</div>';

Ejercicio Bonus: El Reto Git (UD1, UD2, UD3)

Objetivo: Demostrar el flujo de trabajo completo de Git aprendido en las unidades.

Crea un nuevo repositorio local en una carpeta vacía (ej: DAW_Ejercicios).

Commit 1: Completa el Ejercicio 1 (perfil.php). Guárdalo en esa carpeta.

Usa los comandos `git add perfil.php` y `git commit -m "Ejercicio 1: Creada tarjeta de perfil básica"`.

Commit 2: Ahora, crea el Ejercicio 3 (calendario.php) en la misma carpeta.

Usa `git add calendario.php` y `git commit -m "Ejercicio 3: Añadido generador de calendario con 'while'"`.

Push: Crea un repositorio remoto vacío en GitHub (llámalo DAW_Ejercicios_PHP).

Sigue las instrucciones de GitHub para conectar tu repositorio local y subir (`git push`) tus dos commits.

Verifica en la web de GitHub que tus dos ficheros y tus dos commits están allí.