

1. Conceptos básicos: Integración de PHP en HTML. Variables. Operadores

1.1 Integración de PHP en HTML ¡Hola mundo!

Abre el entorno NetBeans y selecciona **Archivo** → **Proyecto Nuevo...** A continuación selecciona **PHP** en el apartado **Categorías** y **Aplicación PHP** en el apartado **Proyectos**. Dale un nombre al proyecto, por ejemplo **Saludo**. Haz clic en **Siguiente** hasta que salga la ventana del editor. Por defecto, se crea un archivo con el nombre `index.php`. Edítalo y escribe el siguiente código:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
  </head>
  <body>
    <?php
      echo "¡Hola mundo!";
    ?>
  </body>
</html>
```

Por último, dale a la tecla **F6** para ejecutar el proyecto.

Como habrás observado, el programa contiene código en HTML mezclado con una sentencia en PHP. Cada vez que quieras insertar código en PHP, deberás encerrarlo entre las etiquetas `<?php` y `?>`. En caso de que todo el código del fichero sea PHP y no haya nada de HTML, se indica únicamente la etiqueta de inicio `<?php`.

La instrucción `echo` sirve para volcar texto en la página HTML. No es necesariamente el texto que se quiere mostrar. Veamos otro ejemplo:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
  </head>
  <body>
    <!-- Utilizo PHP para poner en negrita una palabra -->
    Hola
    <?php echo "<b><u>"; ?>
    mundo
```

```
<?php echo "</u></b>"; ?>
</body>
</html>
```

Observa que esta vez, echo ha servido para volcar en HTML las etiquetas y <u> que hacen que una cadena de caracteres se muestre en negrita y en cursiva respectivamente. Fíjate que después de una sentencia en PHP se escribe un punto y coma.

Aquí tienes otro ejemplo, en este caso mostramos una línea utilizando HTML y otra utilizando PHP:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
  </head>
  <body>
    <!-- Muestra una frase con HTML -->
    Hola mundo<br>
    <!-- Muestra una frase con PHP -->
    <?php echo "Es muy fácil programar en PHP."; ?>
  </body>
</html>
```

1.2 Variables

Definición de variables

Una variable es un contenedor de información, es algo así como una cajita que tiene un nombre y en la que podemos meter un valor. Las variables pueden almacenar números enteros, números decimales, caracteres, cadenas de caracteres (palabras o frases), etc. El contenido de las variables se puede mostrar y se puede cambiar durante la ejecución de una página PHP (por eso se llaman variables).

Los nombres de las variables comienzan con el símbolo del dólar (\$) y no es necesario definirlas como se hace en otros lenguajes de programación como C, Java, Pascal, etc. La misma variable puede contener un número y luego el nombre de una ciudad, no existe restricción en cuanto al tipo como en la mayoría de los lenguajes.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
  </head>
  <body>
    <?php
      $x = 24;
      $pi = 3.1416;
```

```

    $animal = "conejo";
    $saludo = "hola caracola";
    echo $x, "<br>", $pi, "<br>", $animal, "<br>", $saludo;
?>
</body>
</html>

```

En este ejemplo se han definido las variables `$x`, `$pi`, `$animal` y `$saludo`. Con la instrucción `echo` se ha mostrado el valor que contienen, insertando un salto de línea entre ellas. Fíjate que la coma sirve para unir trozos de una cadena de caracteres.

Operadores aritméticos

Los operadores de PHP son similares a los de cualquier otro lenguaje de programación. Estos son los operadores que se pueden aplicar tanto a las variables como a las constantes numéricas:

Operador	Nombre	Ejemplo	Descripción
+	suma	20 + \$x	suma dos números
-	resta	\$a - \$b	resta dos números
*	multiplicación	10 * 7	multiplica dos números
/	división	\$altura / 2	divide dos números
%	módulo	5 % 2	devuelve el resto de la división entera
++	incremento	\$a++	incrementa en 1 el valor de la variable
--	decremento	\$a--	decrementa en 1 el valor de la variable

A continuación se muestra un programa que ilustra el uso de estos operadores:

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
  </head>
  <body>
    <?php
      $a = 8;
      $b = 3;
      echo $a + $b, "<br>";
      echo $a - $b, "<br>";
      echo $a * $b, "<br>";
      echo $a / $b, "<br>";
      $a++;
      echo $a, "<br>";
      $b--;
      echo $b, "<br>";
    ?>
  </body>
</html>

```

```
</body>  
</html>
```

Mientras depuramos un programa, con frecuencia necesitamos ver el valor de las variables. Puedes hacer echo sobre cada una de ellas como hemos visto en los ejemplos anteriores pero es muy cómodo usar

```
print_r(get_defined_vars());
```

que muestra el valor de todas y cada una de las variables que se han definido.

```
<!DOCTYPE html>  
<html>  
  <head>  
    <meta charset="UTF-8">  
  </head>  
  <body>  
    <?php  
      $numero = 20;  
      $palabra = "hola";  
      print_r(get_defined_vars());  
    ?>  
  </body>  
</html>
```

1.3 Ejercicios



Ejercicio 1

Escribe un programa que muestre tu nombre por pantalla. Utiliza código PHP.



Ejercicio 2

Modifica el programa anterior para que muestre tu dirección y tu número de teléfono. Cada dato se debe mostrar en una línea diferente. Mezcla de alguna forma las salidas por pantalla, utilizando tanto HTML como PHP.



Ejercicio 3

Escribe un programa que muestre por pantalla 10 palabras en inglés junto a su correspondiente traducción al castellano. Las palabras deben estar distribuidas en dos columnas. Utiliza la etiqueta `<table>` de HTML.



Ejercicio 4

Escribe un programa que muestre tu horario de clase mediante una tabla. Aunque se puede hacer íntegramente en HTML (igual que los ejercicios anteriores), ve intercalando código HTML y PHP para familiarizarte con éste último.



Ejercicio 5

Escribe un programa que utilice las variables `$x` y `$y`. Asigne los valores 144 y 999 respectivamente. A continuación, muestra por pantalla el valor de cada variable, la suma, la resta, la división y la multiplicación.



Ejercicio 6

Crea la variable `$nombre` y asígnele tu nombre completo. Muestra su valor por pantalla de tal forma que el resultado sea el mismo que el del ejercicio 1.



Ejercicio 7

Crea las variables `$nombre`, `$direccion` y `$telefono` y asígneles los valores adecuados. Muestra los valores por pantalla de tal forma que el resultado sea el mismo que el del ejercicio 2.



Ejercicio 8

Realiza un conversor de euros a pesetas. La cantidad en euros que se quiere convertir deberá estar almacenada en una variable.



Ejercicio 9

Realiza un conversor de pesetas a euros. La cantidad en pesetas que se quiere convertir deberá estar almacenada en una variable.



Ejercicio 10

Escribe un programa que pinte por pantalla una pirámide rellena a base de asteriscos. La base de la pirámide debe estar formada por 9 asteriscos.



Ejercicio 11

Igual que el programa anterior, pero esta vez la pirámide estará hueca (se debe ver únicamente el contorno hecho con asteriscos).



Ejercicio 12

Igual que el programa anterior, pero esta vez la pirámide debe aparecer invertida, con el vértice hacia abajo.