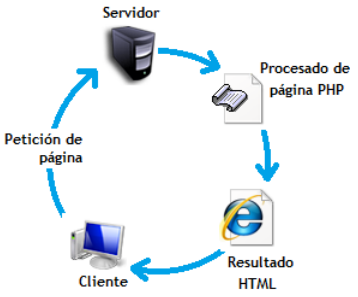
Lenguaje PHP Introducción

Archivos HTML y PHP:

Crear un archivo php es tan sencillo como cambiarle la extensión a un archivo html, por ejemplo podemos pasar de **index.*html*** a ***index.php*** sin ningún inconveniente.

La diferencia está en que ***index.html*** será enviado directamente al cliente que lo solicite e interpretado por un navegador como iexplorer, Firefox, GoogleCrome u otro, mientras que ***index.php*** primero será pasado al intérprete de PHP para que sea procesado y recién entonces se enviará al cliente, y algo muy importante, el html no es un lenguaje de programación, mientras que php si lo es, aunque está orientado exclusivamente al diseño de páginas web.

Petición HTML



Petición php

Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que su navegador lo soporte, es independiente del browser, pero sin embargo para que las páginas PHP funcionen, el servidor donde están alojadas debe soportar PHP.

Bases de la Sintaxis: Inserción de PHP en HTML

El intérprete de php recorre el archivo en busca de sentencias php, estas sentencias se encuentran entre las etiquetas de apertura **<?php** y las de cierre **?>.** En el siguiente ejemplo, el intérprete de php sólo procesará lo que se encuentra entre dichas etiquetas.

```
<html xmlns="http://www.w3.org/1999/xhtml">
   <head>
    <title> new document </title>
   </head>
   <body>
      <?php
         echo "Hola Mundo";
      ?>
   </body>
</html>
```

Separación de instrucciones:

Las instrucciones se separan con ';', en el caso de ser la última instrucción no es necesario el punto y coma.

Comentarios: Los comentarios en PHP pueden ser como en C o C++, /*...*/ ó //

```
<?php
    echo "comentarios"      // comentario en una sola línea
    /*
    Este es un comentario
    de varias líneas
    */
?>
```

Variables:

Son espacios en memoria (*con nombre*) en donde se puede almacenar un determinado tipo de dato, este dato puede cambiar en cualquier parte del programa.

Tipos de Datos:

- *Numéricos:* Pueden ser números enteros o decimales
- *Cadenas de caracteres:* Pueden ser letras, cadenas de letras, palabras o texto largo, en este caso las variables se asignan con una cadena encerrada entre comillas "".
  Nota: Las cadenas pueden estar delimitadas por "" o ''. Si la cadena está delimitada por comillas dobles, cualquier variable incluida dentro de ella será sustituida por su valor.
- *Lógicos:* Admiten valores "Verdadero" (true) o "Falso" (false) únicamente

Conceptos a tener en cuenta en PHP con respecto a las variables:

- Cualquier nombre de variable deberá estar precedido por el símbolo **$**.
- PHP es sensible en cuanto a las mayúsculas y minúsculas. Por ejemplo: $variable, $Variable, $VARIABLE, $VaRiAbLe son 4 variables distintas.
- Los tipos de datos de las variables en PHP no están tan claros como en otros lenguajes, el intérprete asigna el tipo de una variable según el uso que se esté haciendo de ella.
  Por ejemplo:

```
$n="5";          // el intérprete asume que es una cadena
$b=5;            // aquí un dato numérico
$c=true;         // y esta última un dato lógico
```

Operadores en PHP:

Existe un gran número de operadores que se los puede clasificar en:

- De cadena
- Aritméticos
- Lógicos
- De comparación
- De asignación

Operadores de Cadena:

Operador	Descripción
.	Concatenación (une palabras o letras)

Operadores Aritméticos:

Operador	Descripción
+	Suma
-	Resta
*	Producto
/	Cociente
%	Módulo o Resto

Operadores Lógicos:

Operador	Descripción
AND (&&)	"Verdadero" si ambos operandos son "Verdadero"
OR (||)	"Verdadero" si al menos uno de sus operandos es "Verdadero"
NOT (!)	"Verdadero" si su operando es "Falso" "Falso" si su operando es "Verdadero"

Operadores de Comparación:

Operador	Descripción
>	Mayor que...
<	Menor que...
>=	Mayor o Igual que...
<=	Menor o Igual que...
==	Igual a... (no considera el tipo de dato)
===	Igual a... (tiene en cuenta el tipo de dato)
!=	Distinto de...

Operadores de Asignación:

Operador	Descripción
$a = $b	Asigna a $a el contenido de $b
$a += $b	Suma a $a el contenido de $b y lo asigna a $a
$a -= $b	Resta a $a el contenido de $b y lo asigna a $a
$a *= $b	Multiplica $a por $b y lo asigna a $a
$a /= $b	Divide $a por $b y lo asigna a $a
$a .= $b	Añade la cadena $b a la cadena $a y lo asigna a $a

Funciones de escritura en documentos:

Función print() y echo()

la función print es muy fácil de utilizar, inclusive los paréntesis no son necesarios. La estructura es la siguiente:

```
<?php
    print ("una cadena de texto");
    print "con print se muestra el contenido en el navegador";
?>
```

Pero no solamente se puede mostrar texto estático lo que en realidad importa es mostrar el valor de las variables. Veamos, primero declaramos una variable y después la imprimimos con la función print().

```
<?php
    $saludo = "hola";
    print ( "$saludo"); //se imprime la variable saludo
    print "$saludo";
    print $saludo;
?>
```

Lo mismo puede hacerse son la función echo()

```
<?php
    $saludo = "hola";
    echo ("$saludo"); //se imprime la variable saludo
    echo "$saludo";
    echo $saludo;
?>
```

Hasta aquí ambas funciones hacen lo mismo, pero ¿Cuál es la diferencia entre las dos? Bien, una diferencia es que el echo() puede tomar expresiones múltiples, por ejemplo:

```
<?php
    echo "uno", "dos", "tres";
?>
```

Mientras que lo mismo harías con print() utilizando el operador de concatenación "."

```
<?php
    print "uno"."dos"."tres";
?>
```

Otra diferencia es que echo() es más rápido que print()

Ejemplos de Aplicación:

Una página HTML se encuentra estructurada de la siguiente forma:

```
<html>
 <head>
  <title> new document </title>
 </head>
 <body>
  <?php
      // instrucciones php
  ?>
 </body>
</html>
```

Para que sea enviada al intérprete y se ejecuten las sentencias php deberás guardarla con extensión php, por ejemplo: *index.php*

Para ahorrar código y hacer más claro los ejemplos, En adelante usaré sólo las etiquetas de apertura y cierre de php. Tú mantén el resto.

Ejemplo 1- Operadores de Cadena:

```
  <?php
       $a="Hola";
       $b="Mundo";
      $saludo= $a." ".$b; // une Hola con un espacio y con Mundo
       echo $saludo;          // muestra el mensaje "Hola Mundo"
  ?>
```

Ejemplo 2

```
<?php
    $saludo= "Hola Mundo";
    echo "<b>$saludo</b><br />";           // lo muestra en negrita
    echo "<i>".$saludo."</i><br />";              // lo muestra en itálica
    echo "<u>".$saludo."</u><br />";              // lo muestra en subrayado
    echo "<u><i><b>".$saludo."</b></i></u><br />"; // todos juntos
?>
```

Nota 1: La primera salida tiene una variable dentro de la cadena, y como está entre comillas dobles, se reemplaza la variable por su valor.
Nota 2: cada salida se escribe en una línea distinta gracias a que se agregó

Ejemplo 3 - Operadores Aritméticos:

```
<?php
   // cargamos las variables
   $a = 5;
   $b = 10;
   // hacemos los cálculos
   $suma = $a + $b;                // operador de suma
   $resta = $a - $b;               // operador de resta
   $division = $a / $b;   // operador de cociente
   $producto = $a * $b;   // operador de producto
   $resto = $a % $b;               // operador de módulo o resto
   // ahora mostramos los resultados
   echo "La suma es ".$suma."<br />";
   echo "La resta es ".$resta."<br />";
   echo "La división es ".$division."<br />";
   echo "El producto es ".$producto."<br />";
   echo "El resto es ".$resto."<br />";
?>
```

Ejemplo 4 - Podríamos mostrarlo con más detalle

```
<?php
   $a = 5;                // cargamos las variables
   $b = 10;
   $resto = $a % $b;      // hacemos el cálculo y mostramos el resultado
   echo "El resto de dividir $a en $b es ".$resto."<br />";
   echo "ya que tomadas la variables como números enteros<br />";
   echo '$a dividido $b es cero y el resto resulta '.$resto.'.';
?>
```

Nota 1: Nuevamente nota que las variables que se encuentran entre comillas dobles son reemplazadas por sus correspondientes valores.
Nota 1: En la última línea las variables están encerradas entre comillas simples y se muestran como texto normal.

Ejemplo 5- Operadores de asignación:

```
<?php
$textoLargo="esto pretende ser un texto largo y para no perder esta línea ";
$textoLargo.="se está usando un operador de Concatenación y Asignación.<br /><br />";
$textoLargo.="Generalmente se utiliza cuando la cadena es demasiado larga, ";
$textoLargo.='la idea es que a <b>$textoLargo</b> se le va asignando lo que ya trae ';
$textoLargo.="más la nueva cadena que se le agrega, bueno, creo que fue suficiente.";
echo $textoLargo;
?>
```

Nota1: es muy típica su aplicación cuando se hacen consultas largas a la base de datos, para no perder de vista lo que se está consultando.

Ejemplo 6:

```
<?php
    $n1 = 0;
    $n2 = 10;
    $n1 += $n2;            // sumamos a $n1 lo que hay en $n2
    echo $n1."<br />";     // mostramos el resultado.
    $n1 += $n1;            // ahora duplicamos su contenido
    echo $n1;              // y finalmente lo mostramos
?>
```

Ejemplo 7 - Podemos hacer lo mismo con el operador de resta y asignación:

```php
<?php
    $n1=50;
    $n2=1;
    $n1-=$n2;              // restamos de $n1 lo que hay en $n2
    echo $n1."<br />";     // mostramos lo que quedó.
    $n1-=$n1;              // ahora lo llevamos a cero
    echo $n1;              // y finalmente lo mostramos
?>
```

Ejemplo 8 - Podemos hacer lo mismo con el operador de producto y asignación:

```php
<?php
    $n1 = 5;
    $n2 = 4;
    $n1 *= $n2;            // multiplicamos $n1 * $n2 y se lo pasamos a $n1
    echo $n1."<br />";     // mostramos lo que resultó.
    $n1 *= $n1;            // y ahora lo multiplicamos por el mismo número
    echo $n1;              // y finalmente mostramos el inmenso número que quedó.
?>
```

Ejemplo 9 - Podemos hacer lo mismo con el operador de producto y asignación:

```php
<?php
    $n1 = 20;
    $n2 = 4;
    $n1 /= $n2;            // (1) Dividimos $n1 con $n2 y se lo pasamos a $n1
    echo $n1."<br />";     // mostramos lo que resultó.
    $n1 = 20;              // volvemos $n1 a su estado inicial...
    $n1 = $n1 / $n2;       // este sería un equivalente a (1)
    echo $n1;              // mira el resultado...
?>
```