

FUNDAMENTOS DE PHP - 01

Creado en 1995. PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. Existen diferentes versiones

- Cada versión suele incluir novedades

Servidor de despliegue Apache XAMPP.

Definir un proyecto:

–desplegar en C:\xampp\htdocs\1803183\

Nos aseguramos de que el apache (XAMPP) este inicializado

Crear un archivo PHP index.php

– Código web HTML, CSS, JavaScript – Código php, Puede incluir 0-N bloques

Pseudofunción echo:

echo -> imprime código en el documento

Función time();

- Imprime los milisegundos desde el 1 de enero de 1970 (Tiempo Unix)

```
<?php
    echo time();
?>
```

Variables y tipos

- No se declaran variables como tal (no hay palabras reservadas)
 - Se declaran en la primera asignación
 - El tipo se obtiene automáticamente en la asignación
- String (Cadena de texto),
 - Integer (Número entero),
 - Float (número flotante),
 - Boolean (binario: true/false),
 - Array (Colección de elementos),
 - Object (objeto),
 - Null (nulo),
 - Resource (recurso).

- Uso de variable (operador \$).

– \$<nombre de variable>

– Caracteres validos: [A-Z][a-z][0-9] y _

– **Sensibles a mayúsculas y minúsculas**

```
<?php
    echo "Hola Mundo desde VSCode";
    echo time();
    // Comentario - declaraciones
    $variableEntero = 123;
    $variableFlotante = 30.45;
    $variableCadena = "SENA!";
    $variableBoolean = TRUE;
    $saltolinea = "<br />";
    echo nl2br("$variableEntero\n");
    echo $variableFlotante."<br />";
    echo $variableCadena.$saltolinea;
    echo $variableBoolean.$saltolinea;

    $contenido = "PHP Salto de Linea" . PHP_EOL . "La funcion nl2br() "
    . PHP_EOL . "Inserta saltos de línea de HTML";
    $contenido = nl2br($contenido);
    echo $contenido;
?>
```

Operaciones

- Cada tipo de variable admite una serie de operaciones

– Numéricas: operadores aritméticos comunes -, +, *, /, %, **

– Cadenas : . para concatenar

```
<?php
    $variableUno = 1;
    $variableDos = $variableUno + 1;
    $variableTres = "Hola ";
    $variableCuatro = $variableTres . "SENA";
?>
```

Dump de variable

- La función var_dump() muestra la información de la variable

– Muy útil en fase de depuración

```
<?php
    $variableCinco = 1.5;
    var_dump($variableCinco);
    $variableSeis = "1.5";
    var_dump($variableSeis);
?>
```

Estructuras de control, arrays y funciones

Condicional if

- Muy similar a las de otros lenguajes de programación

```
<?php
if ( <expresión> ) {
< sentencias ejecutar si se cumple expresión >
}
?>

<?php
$numero = 4;
if ($numero > 3) {
echo $numero . " Es mayor que 3";
}
?>
```

Operadores de comparación

\$a == \$b	Igual	TRUE si \$a es igual a \$b después de la manipulación de tipos.
\$a === \$b	Idéntico	TRUE si \$a es igual a \$b, y son del mismo tipo.
\$a != \$b	Diferente	TRUE si \$a no es igual a \$b después de la manipulación de tipos.
\$a <> \$b	Diferente	TRUE si \$a no es igual a \$b después de la manipulación de tipos.
\$a !== \$b	No idéntico	TRUE si \$a no es igual a \$b, o si no son del mismo tipo.
\$a < \$b	Menor que	TRUE si \$a es estrictamente menor que \$b.
\$a > \$b	Mayor que	TRUE si \$a es estrictamente mayor que \$b.
\$a <= \$b	Menor o igual que	TRUE si \$a es menor o igual que \$b.
\$a >= \$b	Mayor o igual que	TRUE si \$a es mayor o igual que \$b.

```
<?php
$a= 4;
$b= "4";
if ($a == $b) {
    echo "Mismo valor";
}
if ($a === $b) {
    echo "Mismo valor y mismo tipo";
}
?>
```

Condicional else

- Clausula else, acompaña a la clausula if
- El código se debe ejecutar sí la condición if resulta falsa.

```
<?php
$saleatorio = rand(0, 1000);
echo "Generado: ". $saleatorio . "<br>" ;
if ($saleatorio < 400 || $saleatorio > 800) {
    echo "valor no adecuado, bajo o alto" . "<br>";
} else {
    echo "El valor sí es adecuado" . "<br>";
}
?>
```

Condicional elseif

- Anidar varias condiciones excluyentes
- Primera condición if
- N condiciones elseif
- Condición final else (Se cumple si no se cumple ninguna de las anteriores)

```
<?php
    $temperatura1 = rand(1, 35);
    $temperatura2 = rand(1, 35);
    echo "1: ". $temperatura1 . "<br>";
    echo "2: ". $temperatura2 . "<br>";
    if ($temperatura1 > 20 && $temperatura2 > 20 ) {
        echo "Calor en todos los lados" . "<br>";
    } elseif ($temperatura1 > 20 || $temperatura2 > 20) {
        echo "Calor en algún lado" . "<br>";
    } else {
        echo "Frio!" . "<br>";
    }
?>
```

Estructura Switch

- Incluye varios bloques de condiciones exclusivas
- Tenemos varias alternativas para realizar una acción determinada
- El break provoca que una vez encontrada una coincidencia termine la evaluación de casos

```
<?php
    $dia = rand(0, 3);
    echo "día: " . $dia . "<br>" ;
    switch ($dia) {
        case 1:
            echo "Lunes" . "<br>";
            break;
        case 2:
            echo "Martes" . "<br>";
            break;
        case 3:
            echo "Miércoles" . "<br>";
            break;
        default:
            echo "Día no valido" . "<br>";
    }
?>
```

Repetición while

- While permite repetir código N veces, El código se ejecuta mientras una condición se cumple
- Cuando deja de cumplirse deja de repetirse

```
<?php
    $contar1 = rand(0, 7);
    // Mientras - $contar mayor que 0
    while($contar1 > 0) {
        echo "Contando... : ". $contar1 . "<br>";
        $contar1--; // equivale a "$contar1 = $contar1 - 1";
    }
?>
```

Repetición do while

- Do While es muy similar al While, pero garantiza que **el código se ejecuta al menos una vez**.

– Ejemplo, genera un aleatorio y le va sumando 1 hasta llegar a 4

```
<?php
    $contar2 = rand(0, 7);
    do {
        echo "El número es:" . $contar2 . "<br>";
        $contar2++; // equivale a "$contar2 = $contar2 + 1";
        // Mientras - $contar menor o igual que 4
    } while($contar2 <= 4);
?>
```

Aunque el aleatorio genere un número > 4 el código del do se ejecuta una vez

Repetición for

- Ejecuta un bloque de código de forma repetida
- Normalmente un número de veces indicado
- Utiliza un índice o contador
- Ejemplo ejecutar un código 10 veces
- El índice (variable \$i) va desde 0 hasta 10

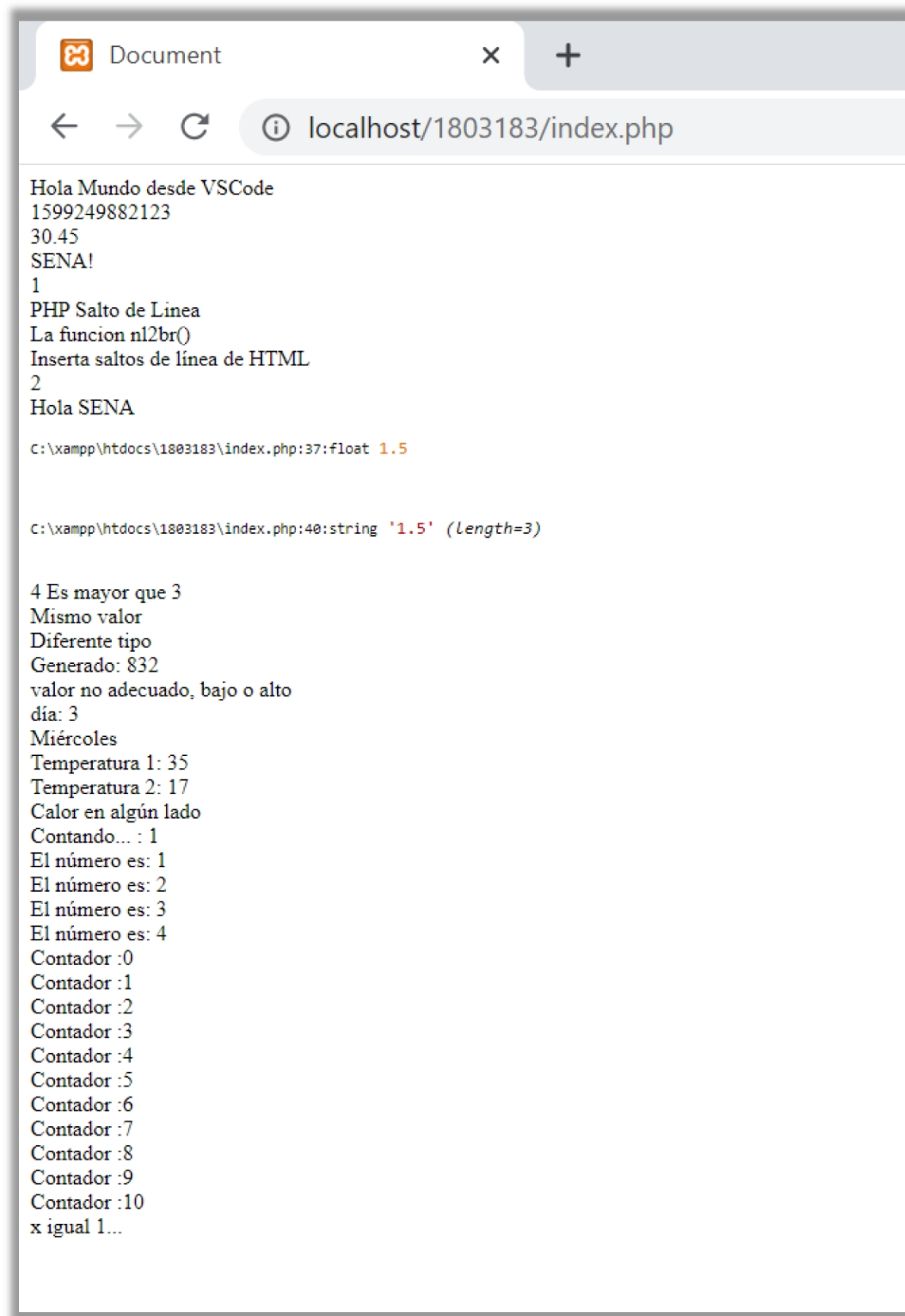
```
<?php
    for ($i = 0; $i <= 10; $i++) {
        echo "Contador :" . $i . "<br>";
    }
?>
```

Sintaxis alternativa de estructuras de control

PHP ofrece una sintaxis alternativa para algunas de sus estructuras de control, como: if, while, for, foreach, y switch. En cada caso, la forma básica de la sintaxis alternativa es cambiar la llave de apertura por dos puntos (:) y la llave de cierre por endif;, endwhile;, endfor;, endforeach;, o endswitch;, respectivamente.

```
<?php
    $x = rand(0, 3);
    if ($x == 1):
        echo "x igual 1";
        echo "...";
    elseif ($x == 2):
        echo "x igual 2";
        echo "!!!";
    else:
        echo "x no es 1 ni 2";
    endif;
?>
```

En el ejemplo anterior, el resultado está anidado dentro de unas sentencias: if, else y elseif escrita en sintaxis alternativa.



```

Hola Mundo desde VSCode
1599249882123
30.45
SENA!
1
PHP Salto de Linea
La funcion nl2br()
Inserta saltos de línea de HTML
2
Hola SENA

C:\xampp\htdocs\1803183\index.php:37:float 1.5

C:\xampp\htdocs\1803183\index.php:40:string '1.5' (length=3)

4 Es mayor que 3
Mismo valor
Diferente tipo
Generado: 832
valor no adecuado, bajo o alto
día: 3
Miércoles
Temperatura 1: 35
Temperatura 2: 17
Calor en algún lado
Contando... : 1
El número es: 1
El número es: 2
El número es: 3
El número es: 4
Contador :0
Contador :1
Contador :2
Contador :3
Contador :4
Contador :5
Contador :6
Contador :7
Contador :8
Contador :9
Contador :10
x igual 1...
    
```