

## **DIPLOMADO: PROGRAMACIÓN EN PHP**

### **OPERADORES**

Los operadores se utilizan para realizar operaciones, valga la redundancia, en variables y valores. PHP divide los operadores en los siguientes grupos:

- Operadores aritméticos
- Operadores de asignación
- Operadores de comparación
- Operadores de incremento / decremento
- Operadores de cadenas de texto
- Operador ternario

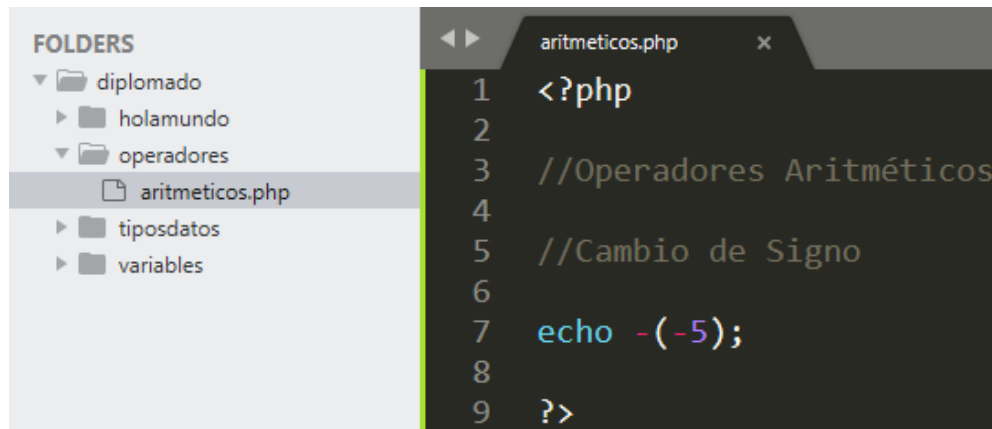
#### **Operadores aritméticos**

¿Recuerda la aritmética básica de la escuela? Estos funcionan igual que aquellos. Los operadores aritméticos de PHP se utilizan con valores numéricos para realizar operaciones aritméticas comunes, como la suma, resta, multiplicación, etc.

Operador	Descripción	Ejemplo en Valores	Resultado
-	Cambio de Signo	$-(-5)$	5
+	Suma	$5 + 8$	13
-	Resta	$3 - 7$	-4
*	Multiplicación	$5 * 4$	20
/	División	$18 / 3$	6
%	Módulo	$50 \% 2$	0
**	Exponenciación	$6 ** 4$	1296

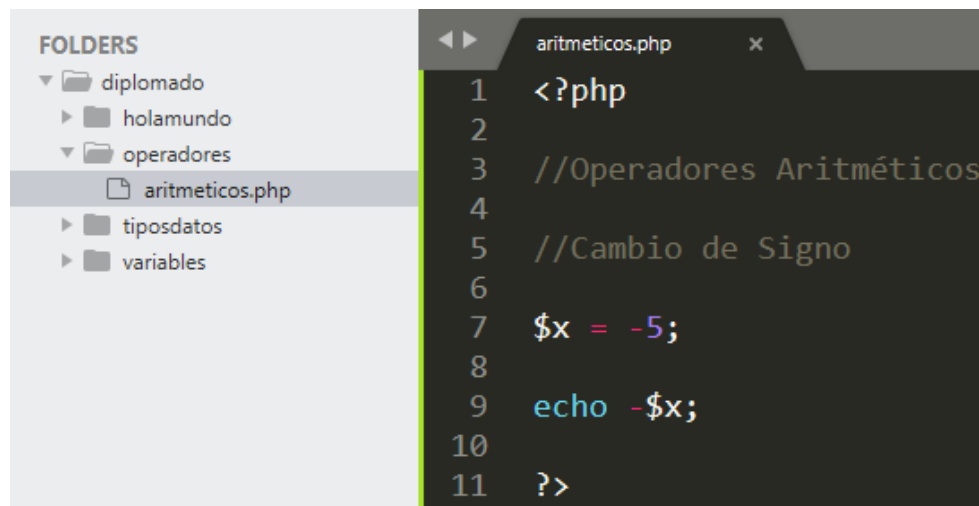
## Operador de Cambio de Signo

El operador de cambio de signo, como su mismo nombre lo indica, permite realizar un cambio de signo (negativo o positivo) a una variable o un valor en específico.



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a directory structure with folders 'diplomado', 'holamundo', 'operadores', 'tiposdatos', and 'variables'. The 'operadores' folder is expanded, and 'aritmeticos.php' is selected. The code editor shows the following code:

```
1 <?php
2
3 //Operadores Aritméticos
4
5 //Cambio de Signo
6
7 echo -(-5);
8
9 ?>
```

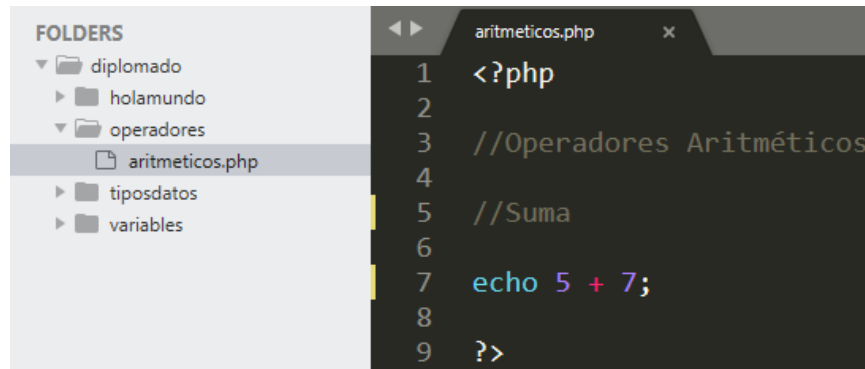


The screenshot shows the same code editor with the same file explorer. The code editor now shows the following code:

```
1 <?php
2
3 //Operadores Aritméticos
4
5 //Cambio de Signo
6
7 $x = -5;
8
9 echo -$x;
10
11 ?>
```

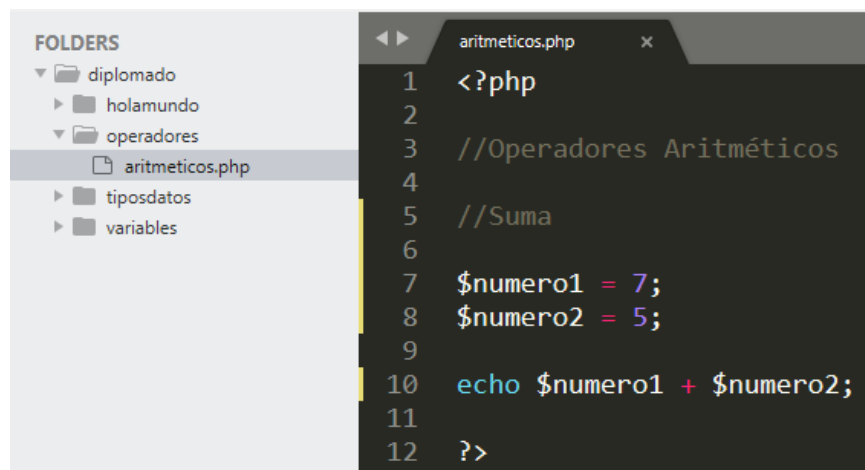
## Operador de Suma

El operador de suma (+), como su mismo nombre lo indica, permite realizar una suma sobre dos o X cantidad de variables o valores en específico.



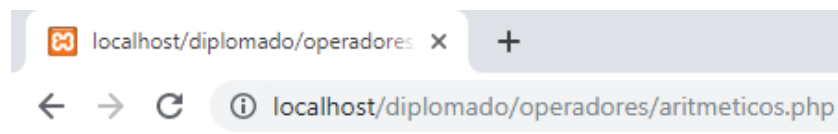
The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a directory structure: diplomado > holamundo > operadores > aritmeticos.php. The code editor shows the following PHP code:

```
1 <?php
2
3 //Operadores Aritméticos
4
5 //Suma
6
7 echo 5 + 7;
8
9 ?>
```



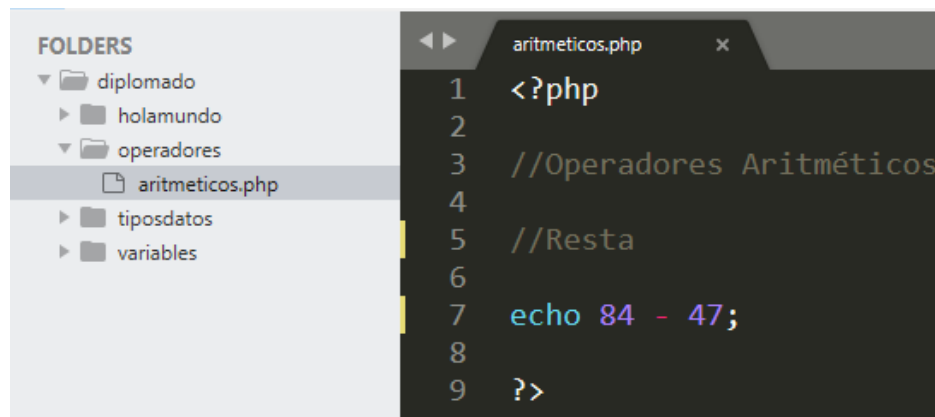
The screenshot shows the same code editor with the same file explorer. The code editor now shows the following PHP code:

```
1 <?php
2
3 //Operadores Aritméticos
4
5 //Suma
6
7 $numero1 = 7;
8 $numero2 = 5;
9
10 echo $numero1 + $numero2;
11
12 ?>
```



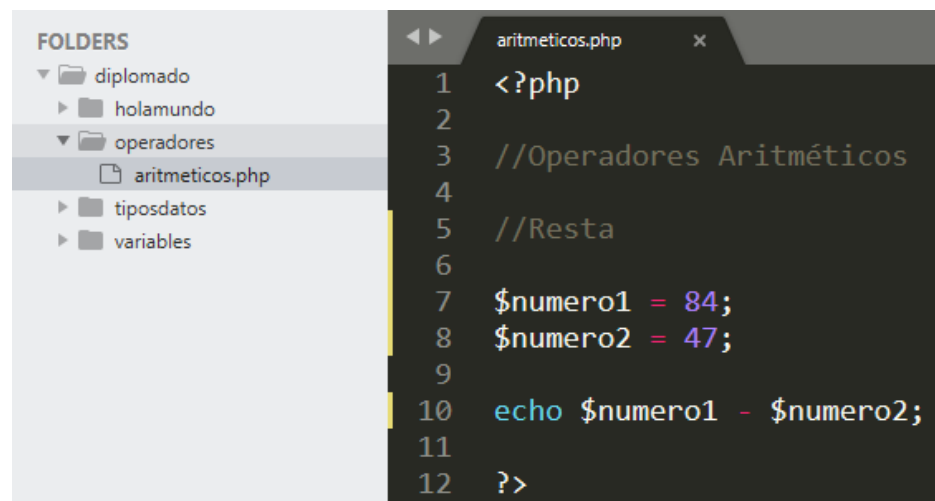
## Operador de Resta

El operador de resta (-), como su mismo nombre lo indica, permite realizar una resta sobre dos o X cantidad de variables o valores en específico.



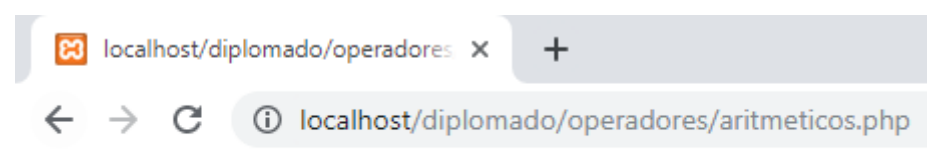
The screenshot shows a code editor with a file named 'aritmeticos.php'. The left sidebar shows a folder structure: 'diplomado' containing 'holamundo', 'operadores' (containing 'aritmeticos.php'), 'tiposdatos', and 'variables'. The code in the editor is as follows:

```
1 <?php
2
3 //Operadores Aritméticos
4
5 //Resta
6
7 echo 84 - 47;
8
9 ?>
```



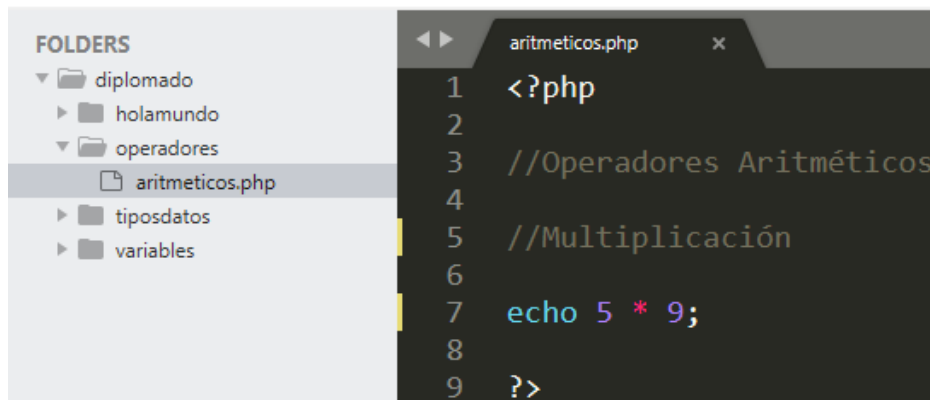
The screenshot shows the same code editor with 'aritmeticos.php'. The code has been updated to use variables for the numbers:

```
1 <?php
2
3 //Operadores Aritméticos
4
5 //Resta
6
7 $numero1 = 84;
8 $numero2 = 47;
9
10 echo $numero1 - $numero2;
11
12 ?>
```



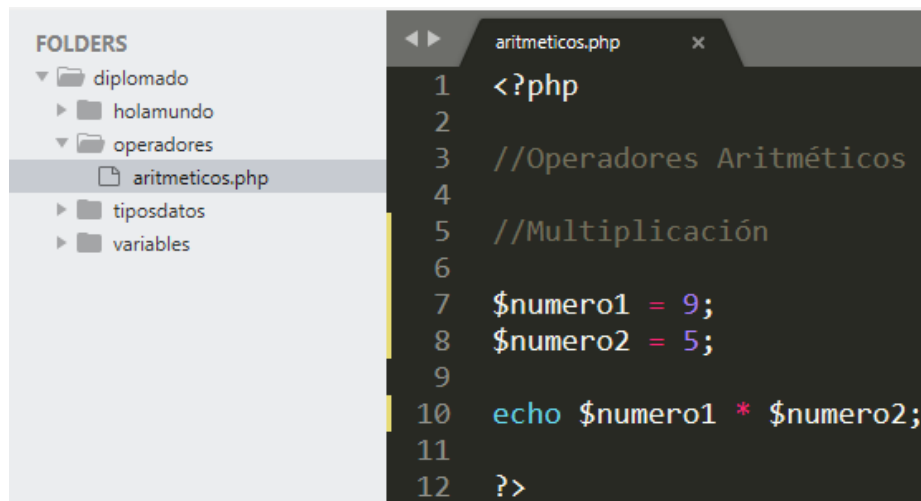
## Operador de Multiplicación

El operador de multiplicación (\*), como su mismo nombre lo indica, permite realizar una multiplicación sobre dos o X cantidad de variables o valores en específico.



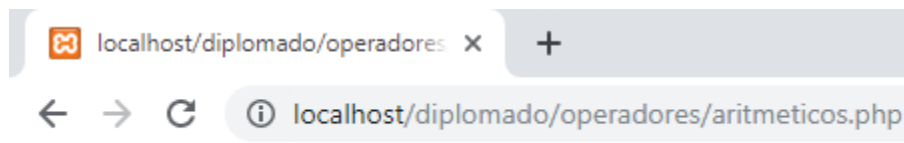
The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a folder structure: diplomado > operadores > aritmeticos.php. The code editor shows the following PHP code:

```
1 <?php
2
3 //Operadores Aritméticos
4
5 //Multiplicación
6
7 echo 5 * 9;
8
9 ?>
```



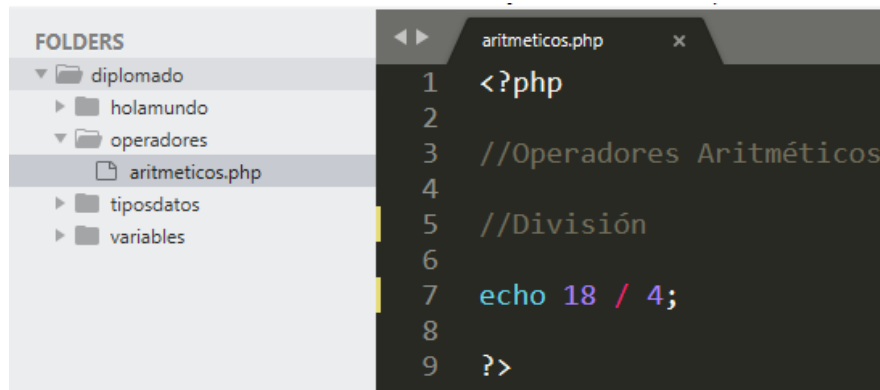
The screenshot shows the same code editor with the same file structure. The code editor now shows the following PHP code:

```
1 <?php
2
3 //Operadores Aritméticos
4
5 //Multiplicación
6
7 $numero1 = 9;
8 $numero2 = 5;
9
10 echo $numero1 * $numero2;
11
12 ?>
```

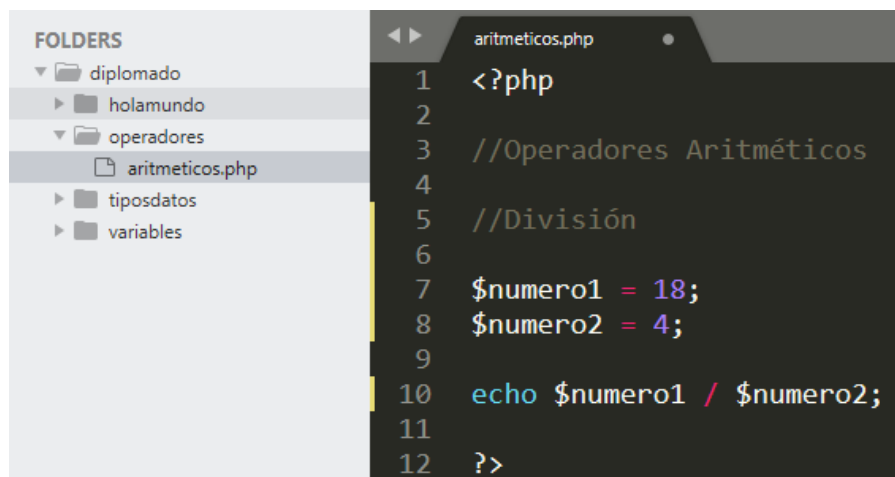


## Operador de División

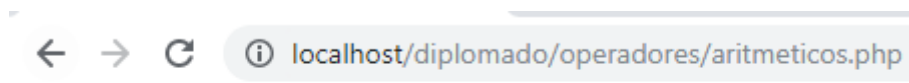
El operador de división (/), como su mismo nombre lo indica, permite realizar una división sobre dos o X cantidad de variables o valores en específico.



```
1 <?php
2
3 //Operadores Aritméticos
4
5 //División
6
7 echo 18 / 4;
8
9 ?>
```

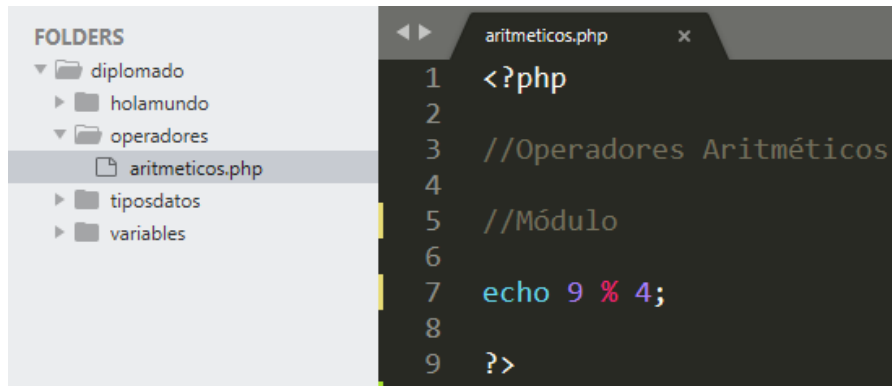


```
1 <?php
2
3 //Operadores Aritméticos
4
5 //División
6
7 $numero1 = 18;
8 $numero2 = 4;
9
10 echo $numero1 / $numero2;
11
12 ?>
```



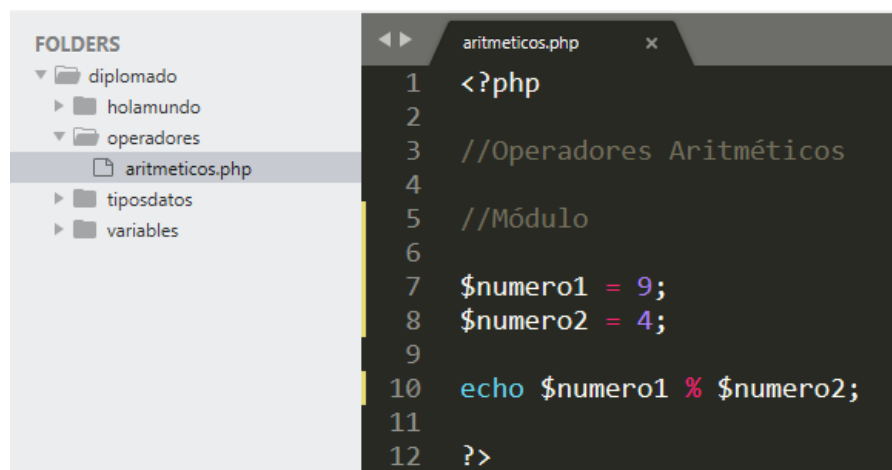
## Operador de Módulo

El operador de módulo (%), permite hallar el residuo de una división a partir de dos variables o valores.



The screenshot shows a code editor with a file named 'aritmeticos.php'. The left sidebar shows a folder structure: 'diplomado' > 'operadores' > 'aritmeticos.php'. The code in the editor is as follows:

```
1 <?php
2
3 //Operadores Aritméticos
4
5 //Módulo
6
7 echo 9 % 4;
8
9 ?>
```



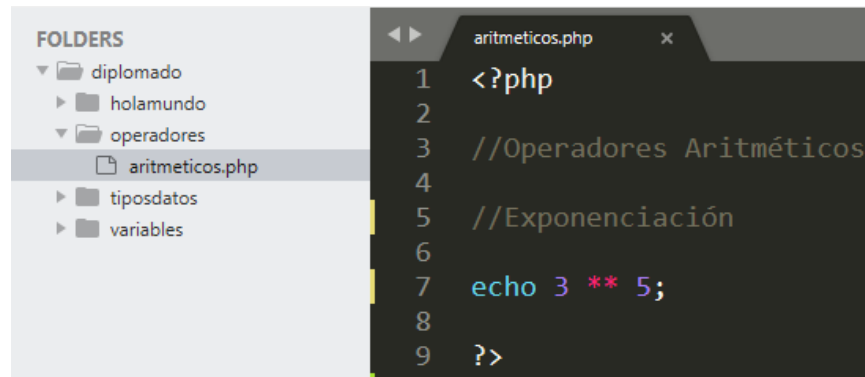
The screenshot shows the same code editor with 'aritmeticos.php'. The code has been updated to use variables for the modulo operation:

```
1 <?php
2
3 //Operadores Aritméticos
4
5 //Módulo
6
7 $numero1 = 9;
8 $numero2 = 4;
9
10 echo $numero1 % $numero2;
11
12 ?>
```

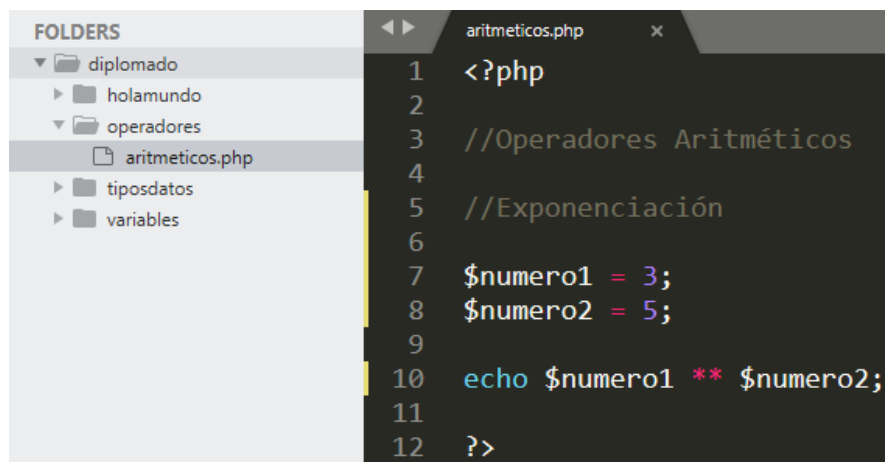
← → ↻ ⓘ localhost/diplomado/operadores/aritmeticos.php

## Operador Exponencial

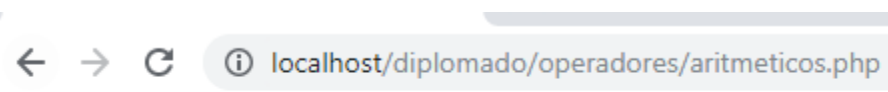
El operador de exponencial (\*\*), permite elevar la variable o el valor de la izquierda, cuantas veces la variable o el valor de la derecha lo establezca.



```
1 <?php
2
3 //Operadores Aritméticos
4
5 //Exponenciación
6
7 echo 3 ** 5;
8
9 ?>
```



```
1 <?php
2
3 //Operadores Aritméticos
4
5 //Exponenciación
6
7 $numero1 = 3;
8 $numero2 = 5;
9
10 echo $numero1 ** $numero2;
11
12 ?>
```





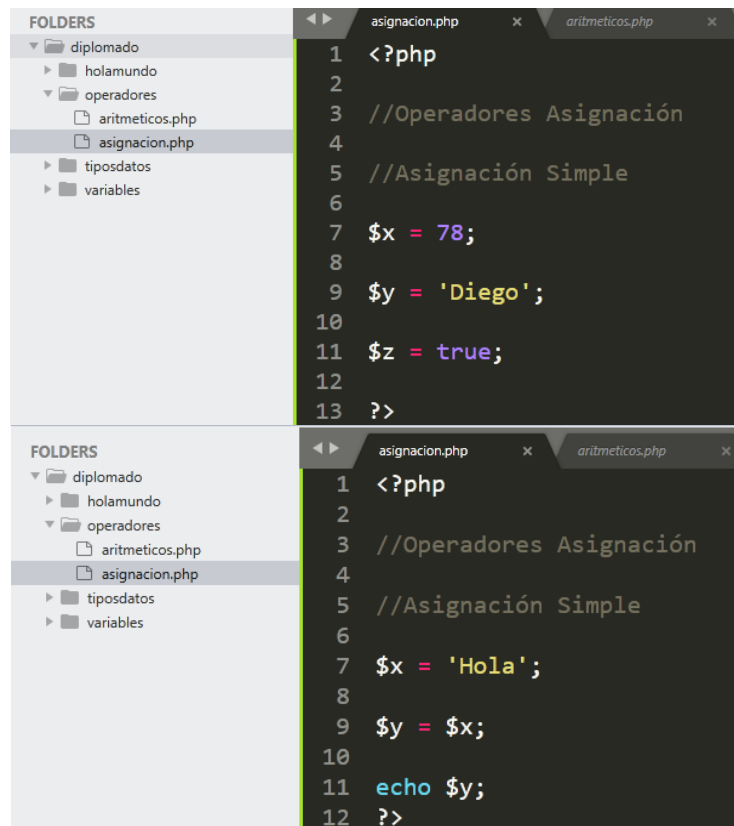
## Operadores de Asignación

Los operadores de asignación de PHP se utilizan con valores numéricos para escribir un valor en una variable. El operador de asignación básica en PHP es "=". Esto significa que el operando de la izquierda se establece en el valor de la expresión de asignación de la derecha.

Operación	Equivalencia	Descripción
$\$x = y$	$\$x = y$	El operando de la izquierda se establece en el valor de la expresión de la derecha
$\$x += y$	$\$x = \$x + \$y$	Adición
$\$x -= y$	$\$x = \$x - \$y$	Sustracción
$\$x *= y$	$\$x = \$x * \$y$	Multiplicación
$\$x /= y$	$\$x = \$x / y$	División
$\$x \% = y$	$\$x = \$x \% \$y$	Módulo

## Asignación Simple

En la asignación simple, la variable de la izquierda toma el valor o variable de la derecha, inmediatamente después del signo igual "=".

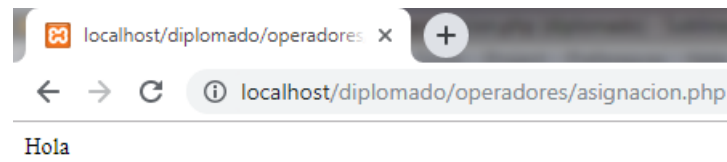


```

1 <?php
2
3 //Operadores Asignación
4
5 //Asignación Simple
6
7 $x = 78;
8
9 $y = 'Diego';
10
11 $z = true;
12
13 ?>
  
```

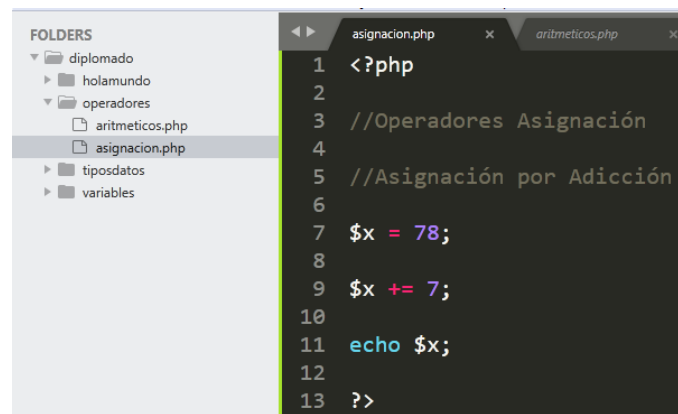
```

1 <?php
2
3 //Operadores Asignación
4
5 //Asignación Simple
6
7 $x = 'Hola';
8
9 $y = $x;
10
11 echo $y;
12 ?>
  
```

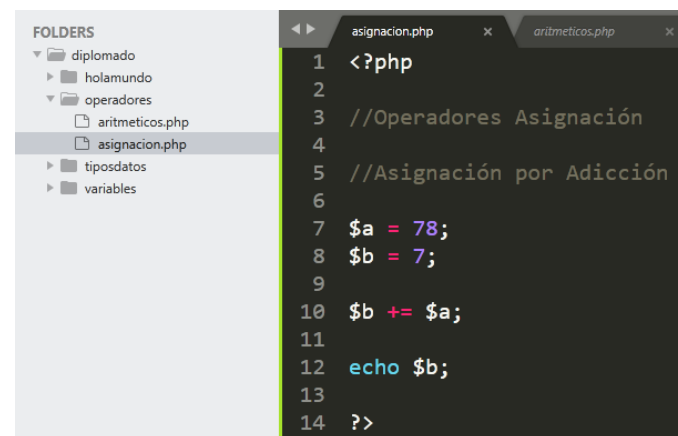


## Asignación por Adición

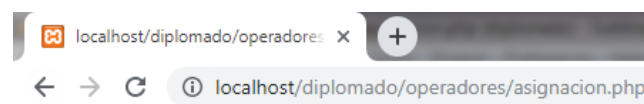
El operador de asignación por adición agrega el valor del operando derecho a la variable y le asigna el resultado a la misma.



```
1 <?php
2
3 //Operadores Asignación
4
5 //Asignación por Adicción
6
7 $x = 78;
8
9 $x += 7;
10
11 echo $x;
12
13 ?>
```

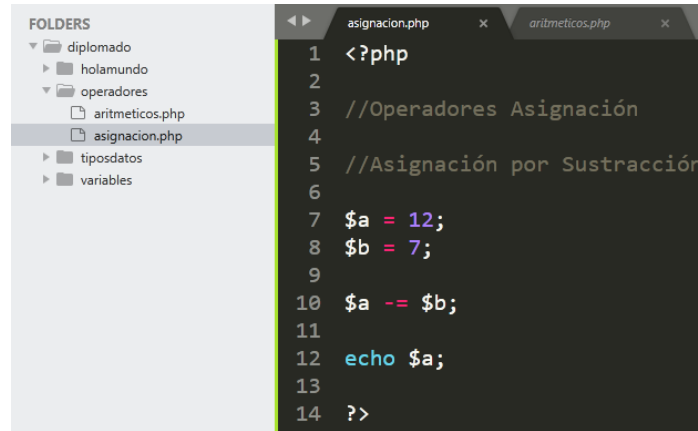


```
1 <?php
2
3 //Operadores Asignación
4
5 //Asignación por Adicción
6
7 $a = 78;
8 $b = 7;
9
10 $b += $a;
11
12 echo $b;
13
14 ?>
```

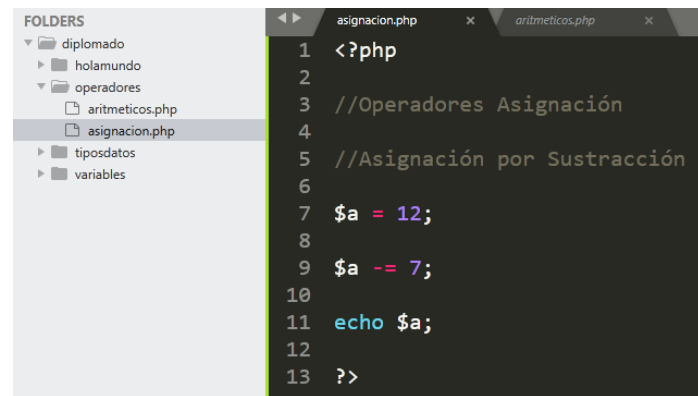


## Asignación por Sustracción

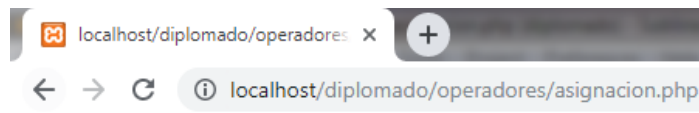
El operador de asignación por sustracción, sustrae el valor del operador derecho desde la variable y asigna el resultado a la misma.



```
1 <?php
2
3 //Operadores Asignación
4
5 //Asignación por Sustracción
6
7 $a = 12;
8 $b = 7;
9
10 $a -= $b;
11
12 echo $a;
13
14 ?>
```

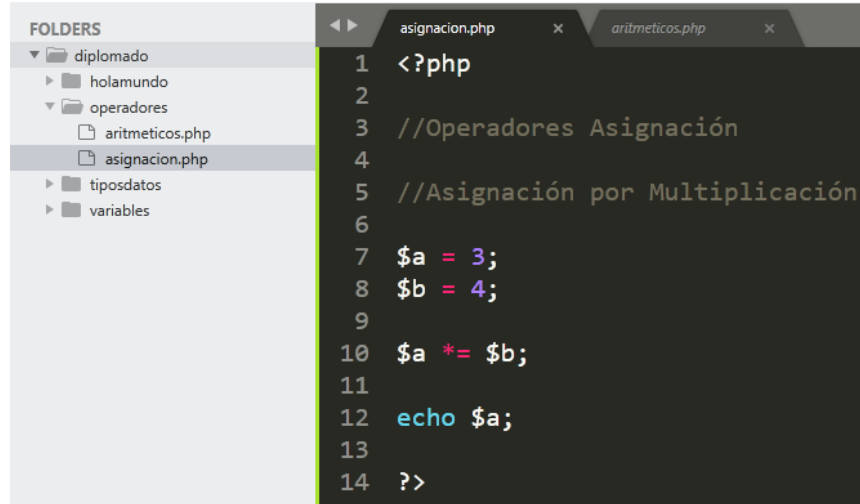


```
1 <?php
2
3 //Operadores Asignación
4
5 //Asignación por Sustracción
6
7 $a = 12;
8
9 $a -= 7;
10
11 echo $a;
12
13 ?>
```

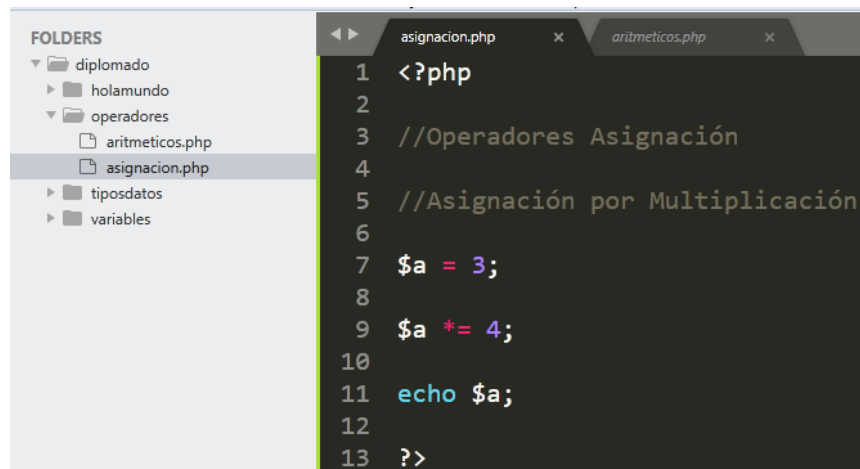


## Asignación por Multiplicación

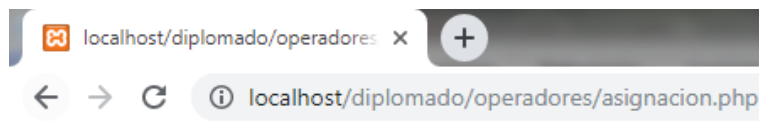
El operador de asignación por multiplicación, multiplica la variable por el valor del operando derecho y asigna el resultado a la misma.



```
1 <?php
2
3 //Operadores Asignación
4
5 //Asignación por Multiplicación
6
7 $a = 3;
8 $b = 4;
9
10 $a *= $b;
11
12 echo $a;
13
14 ?>
```

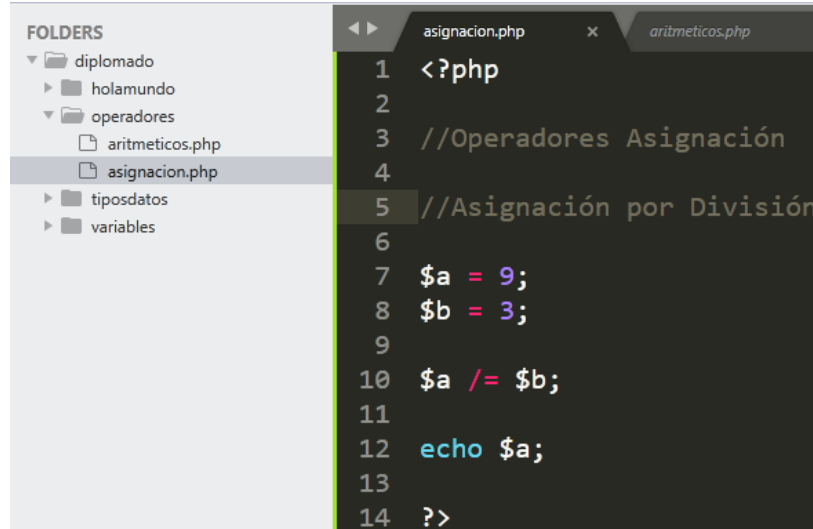


```
1 <?php
2
3 //Operadores Asignación
4
5 //Asignación por Multiplicación
6
7 $a = 3;
8
9 $a *= 4;
10
11 echo $a;
12
13 ?>
```

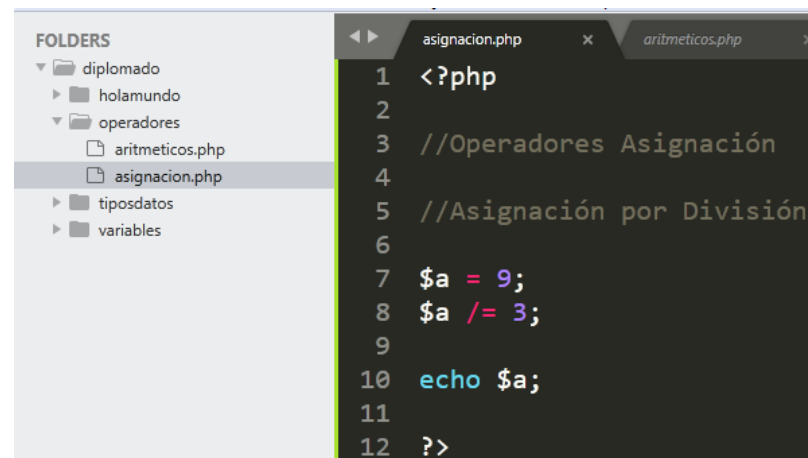


## Asignación por División

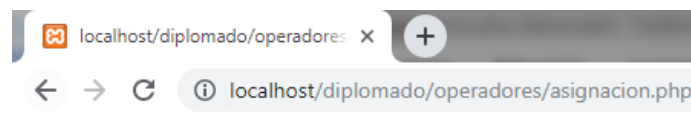
El operador de asignación por división, divide la variable por el valor del operando derecho y asigna el resultado a la misma.



```
1 <?php
2
3 //Operadores Asignación
4
5 //Asignación por División
6
7 $a = 9;
8 $b = 3;
9
10 $a /= $b;
11
12 echo $a;
13
14 ?>
```

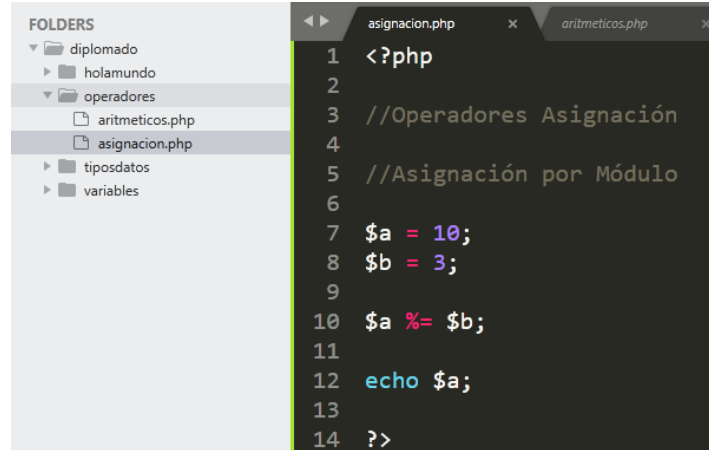


```
1 <?php
2
3 //Operadores Asignación
4
5 //Asignación por División
6
7 $a = 9;
8 $a /= 3;
9
10 echo $a;
11
12 ?>
```

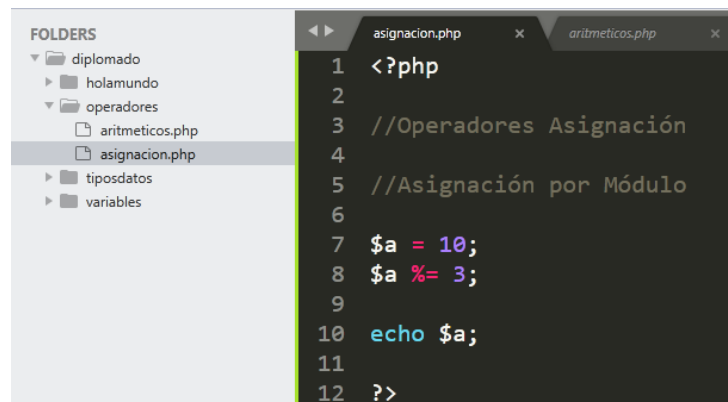


## Asignación por Módulo

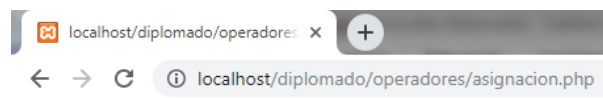
El operador de asignación por módulo (residuo) divide la variable por el valor del operador derecho y asigna el residuo a la misma.



```
1 <?php
2
3 //Operadores Asignación
4
5 //Asignación por Módulo
6
7 $a = 10;
8 $b = 3;
9
10 $a %= $b;
11
12 echo $a;
13
14 ?>
```



```
1 <?php
2
3 //Operadores Asignación
4
5 //Asignación por Módulo
6
7 $a = 10;
8 $a %= 3;
9
10 echo $a;
11
12 ?>
```



## Operadores de Comparación

Los operadores de comparación, como su nombre lo indica, permiten comparar dos valores, sin importar el tipo, objetos, cadenas de texto, números, entre otros; obteniendo siempre un resultado, verdadero o falso, al ser un condicional.

Entre los principales operadores de comparación se encuentran los siguientes:

Operación	Nombre	Operación
==	Igualdad	$x == y$
===	Idéntico	$x === y$
!=	Diferente	$x != y$
<>	Diferente	$x <> y$
!==	No idéntico	$x !== y$
>	Mayor qué	$x > y$
<	Menor qué	$x < y$
>=	Mayor o igual qué	$x >= y$
<=	Menor o igual qué	$x <= y$

## Operador de Igualdad

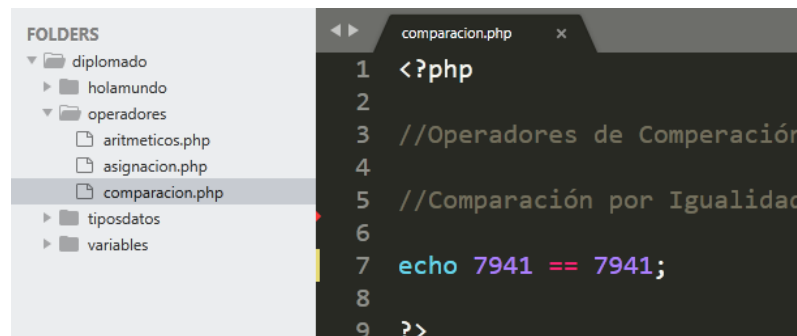
El operador de igual toma ambos valores o variables y compara que sean iguales. En caso de que lo sea, el resultado será true (verdadero) o en caso contrario false (falso).

### Haciendo uso del operador de igualdad ==

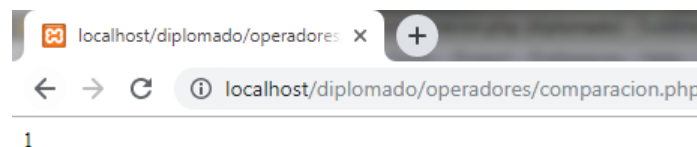
#### Caso verdadero



```
1 <?php
2
3 //Operadores de Comperación
4
5 //Comparación por Igualdad
6
7 $numero1 = 7941;
8
9 $numero2 = 7941;
10
11 echo $numero1 == $numero2;
12
13 ?>
```



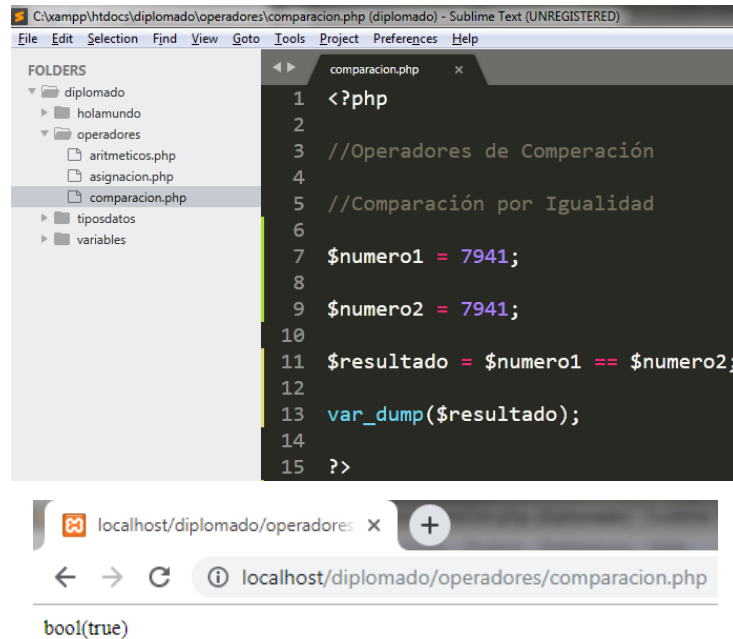
```
1 <?php
2
3 //Operadores de Comperación
4
5 //Comparación por Igualdad
6
7 echo 7941 == 7941;
8
9 ?>
```



En PHP algunas ocasiones el resultado true (verdadero) se ve traducido en 1, que es el equivalente al valor ya mencionado y opera la misma forma. Si se desea conocer el true (verdadero) o ideal es almacenar el resultado de la



operación en una variable y realizar un var\_dump para obtener el resultado esperado.



```
1 <?php
2
3 //Operadores de Comperación
4
5 //Comparación por Igualdad
6
7 $numero1 = 7941;
8
9 $numero2 = 7941;
10
11 $resultado = $numero1 == $numero2;
12
13 var_dump($resultado);
14
15 ?>
```

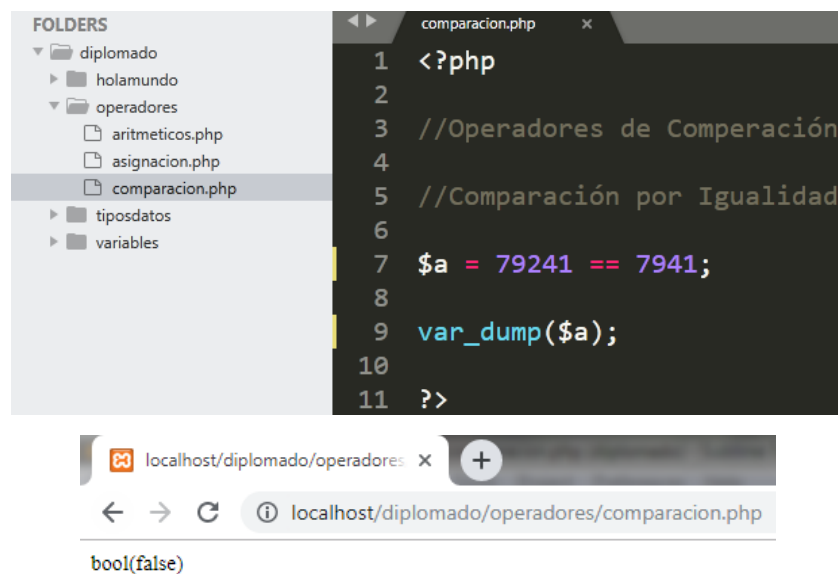
localhost/diplomado/operadores x +

localhost/diplomado/operadores/comparacion.php

bool(true)

## Caso falso

En caso contrario para obtener el resultado false (falso), el procedimiento es el mismo.



```
1 <?php
2
3 //Operadores de Comperación
4
5 //Comparación por Igualdad
6
7 $a = 79241 == 7941;
8
9 var_dump($a);
10
11 ?>
```

localhost/diplomado/operadores x +

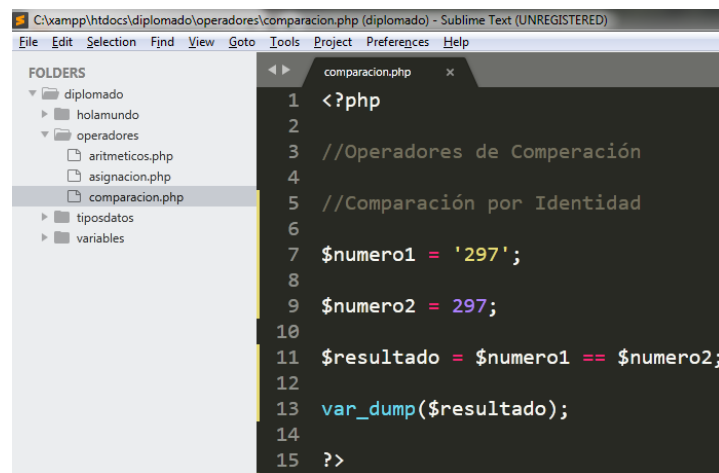
localhost/diplomado/operadores/comparacion.php

bool(false)

## Operador de Identidad

El operador de identidad toma ambos valores o variables y compara que sean iguales, pero adicional a la comparación de valores, el operador de identidad también compara la igualdad de tipos. En caso de que lo sea, el resultado será true (verdadero) o en caso contrario false (falso).

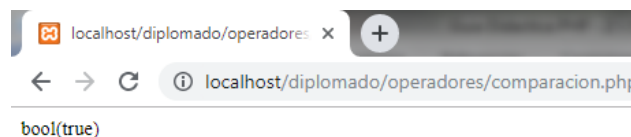
En el siguiente ejemplo se realiza una igualdad con valores similares **297**, pero tipos diferentes: un string y un int.



```
C:\xampp\htdocs\diplomado\operadores\comparacion.php (diplomado) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS
└─ diplomado
  └─ holamundo
    └─ operadores
      └─ aritmeticos.php
      └─ asignacion.php
      └─ comparacion.php
      └─ tiposdatos
      └─ variables

comparacion.php x
1 <?php
2
3 //Operadores de Comparación
4
5 //Comparación por Identidad
6
7 $numero1 = '297';
8
9 $numero2 = 297;
10
11 $resultado = $numero1 == $numero2;
12
13 var_dump($resultado);
14
15 ?>
```

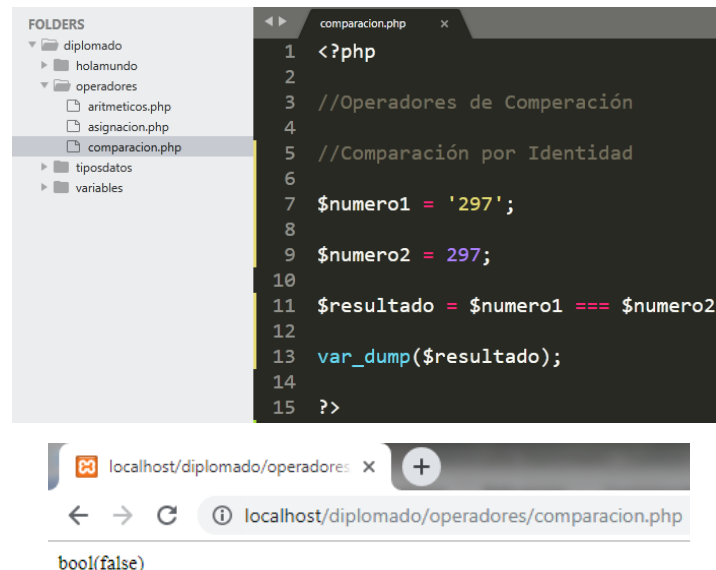


El resultado es verdadero dado que la igualdad solo comparará valores e ignorará los tipos, el operador de identidad soluciona estos casos de la siguiente forma.

## Haciendo uso del operador de identidad ===

### Caso falso

En caso false (falso) aplicando el operador al ejemplo anterior el resultado será el siguiente, téngase en cuenta que la comparación ahora es por tipo y valor.



```
1 <?php
2
3 //Operadores de Comperación
4
5 //Comparación por Identidad
6
7 $numero1 = '297';
8
9 $numero2 = 297;
10
11 $resultado = $numero1 === $numero2;
12
13 var_dump($resultado);
14
15 ?>
```

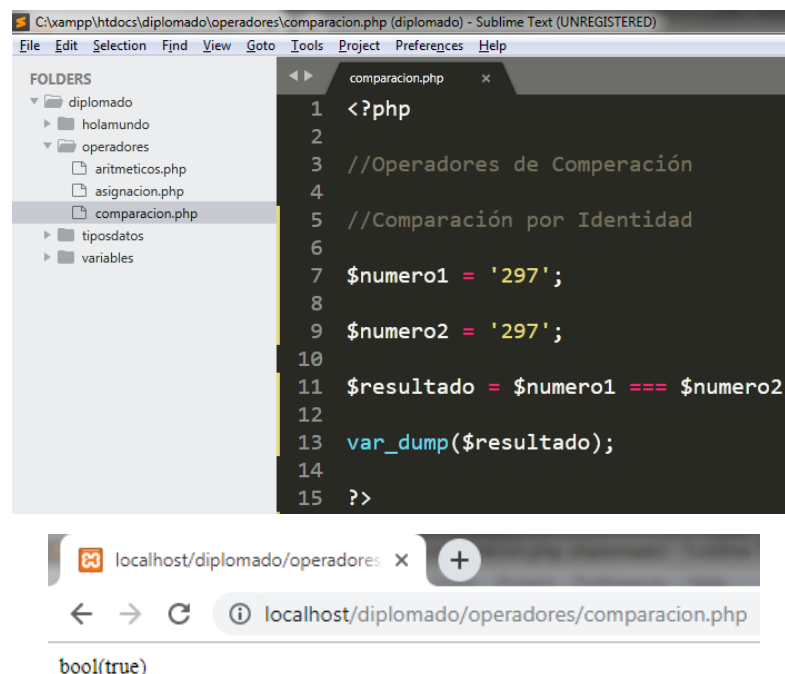
localhost/diplomado/operadores x +

localhost/diplomado/operadores/comparacion.php

bool(false)

### Caso verdadero

Un ejemplo el resultado true (verdadero) en una comparación por identidad, será uno donde valor y tipo sean iguales, como en el siguiente caso:



```
1 <?php
2
3 //Operadores de Comperación
4
5 //Comparación por Identidad
6
7 $numero1 = '297';
8
9 $numero2 = '297';
10
11 $resultado = $numero1 === $numero2;
12
13 var_dump($resultado);
14
15 ?>
```

localhost/diplomado/operadores x +

localhost/diplomado/operadores/comparacion.php

bool(true)

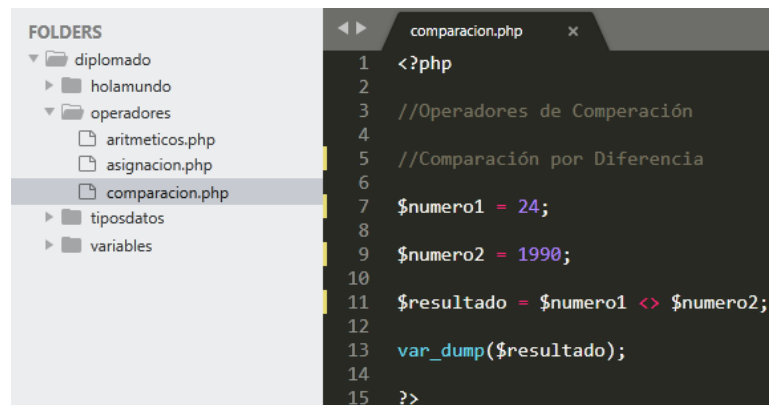
## Operadores de diferencia

El operador de diferencia toma ambos valores o variables y compara que no sean iguales (es decir, diferentes). En caso de que lo sea, el resultado será true (verdadero) o en caso contrario false (falso).

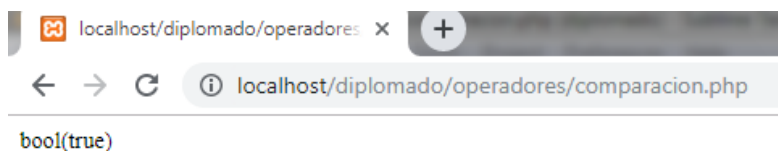
El operador de diferencia cuenta con dos variantes que tienen el mismo resultado: `<>` y `!=`

## Haciendo uso del operador `<>`

### Caso verdadero



```
1 <?php
2
3 //Operadores de Comparación
4
5 //Comparación por Diferencia
6
7 $numero1 = 24;
8
9 $numero2 = 1990;
10
11 $resultado = $numero1 <> $numero2;
12
13 var_dump($resultado);
14
15 ?>
```

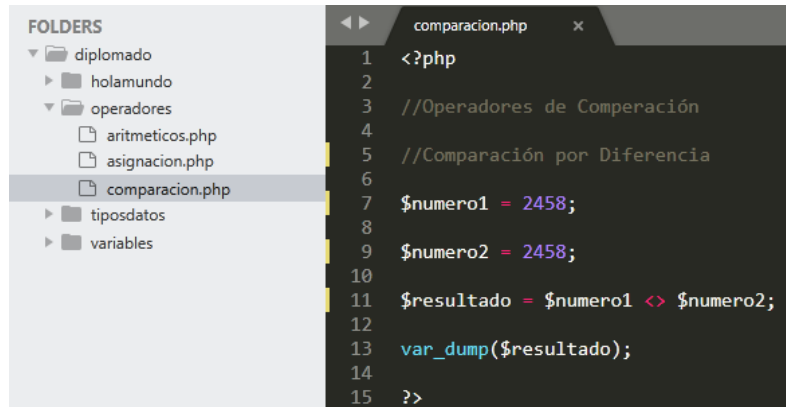


localhost/diplomado/operadores x +

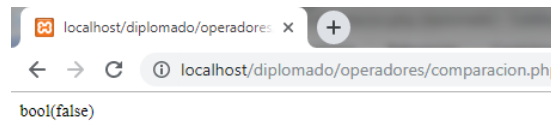
localhost/diplomado/operadores/comparacion.php

bool(true)

## Caso falso



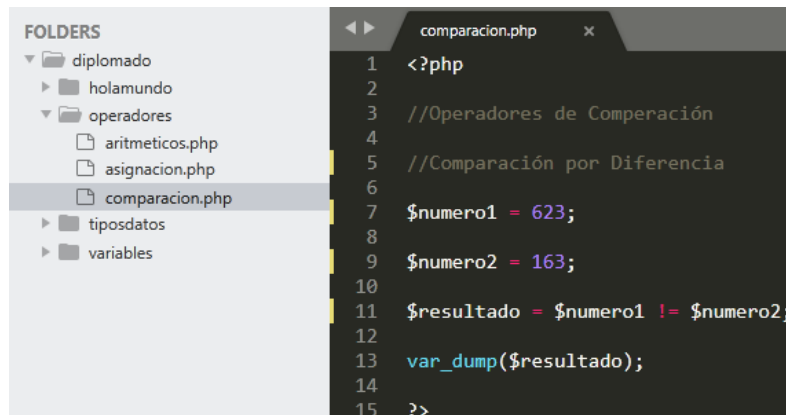
```
1 <?php
2
3 //Operadores de Comparación
4
5 //Comparación por Diferencia
6
7 $numero1 = 2458;
8
9 $numero2 = 2458;
10
11 $resultado = $numero1 <> $numero2;
12
13 var_dump($resultado);
14
15 ?>
```



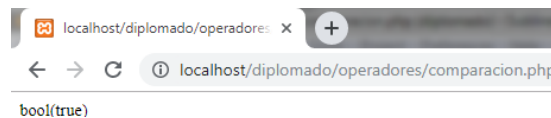
localhost/diplomado/operadores: x +  
localhost/diplomado/operadores/comparacion.php  
bool(false)

## Haciendo uso del operador !=

## Caso verdadero

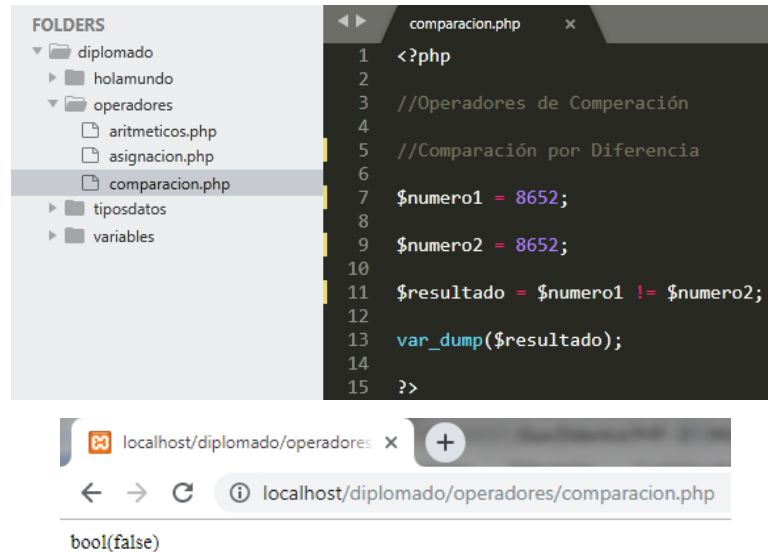


```
1 <?php
2
3 //Operadores de Comparación
4
5 //Comparación por Diferencia
6
7 $numero1 = 623;
8
9 $numero2 = 163;
10
11 $resultado = $numero1 != $numero2;
12
13 var_dump($resultado);
14
15 ?>
```



localhost/diplomado/operadores: x +  
localhost/diplomado/operadores/comparacion.php  
bool(true)

## Caso falso



The screenshot shows a code editor with a file named `comparacion.php`. The code is as follows:

```
1 <?php
2
3 //Operadores de Comperación
4
5 //Comparación por Diferencia
6
7 $numero1 = 8652;
8
9 $numero2 = 8652;
10
11 $resultado = $numero1 != $numero2;
12
13 var_dump($resultado);
14
15 ?>
```

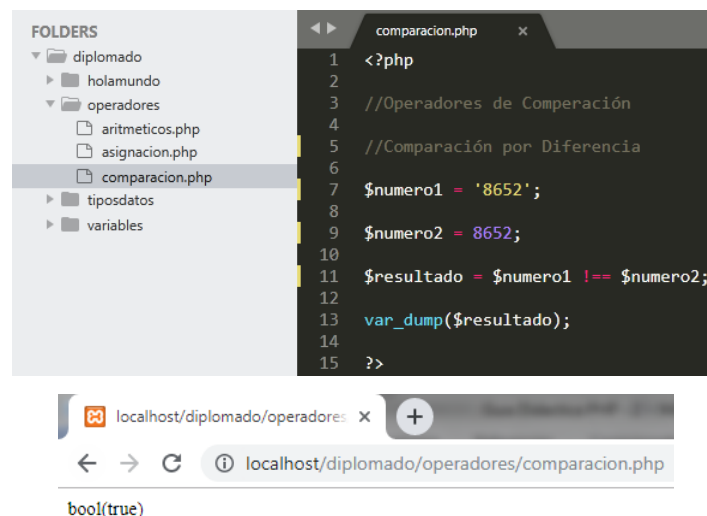
Below the code editor, a browser window is shown with the URL `localhost/diplomado/operadores/comparacion.php`. The output displayed in the browser is `bool(false)`.

## Operador No Idéntico

El operador no idéntico es el equivalente al operador de identidad, pero aplicado a resultados de diferencia, toma ambos valores o variables y compara que no sean iguales (es decir, diferentes) en cuanto a valor y tipo de dato. En caso de que lo sea, el resultado será true (verdadero) o en caso contrario false (falso).

## Haciendo uso del operador no idéntico !=

### Caso verdadero

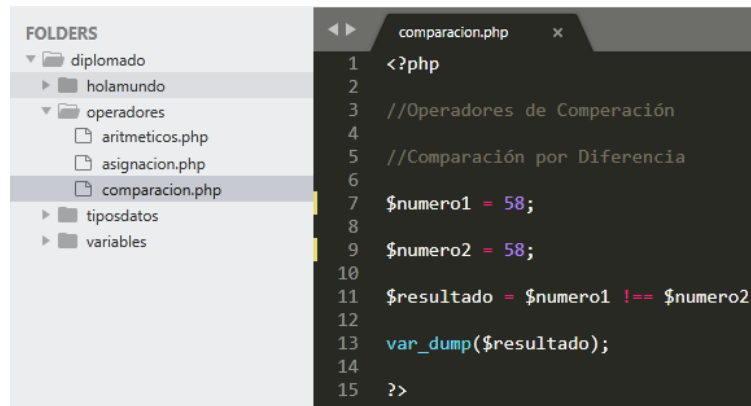


The screenshot shows a code editor with a file named `comparacion.php`. The code is as follows:

```
1 <?php
2
3 //Operadores de Comperación
4
5 //Comparación por Diferencia
6
7 $numero1 = '8652';
8
9 $numero2 = 8652;
10
11 $resultado = $numero1 != $numero2;
12
13 var_dump($resultado);
14
15 ?>
```

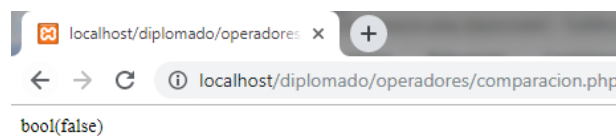
Below the code editor, a browser window is shown with the URL `localhost/diplomado/operadores/comparacion.php`. The output displayed in the browser is `bool(true)`.

## Caso falso



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a folder structure: diplomado > holamundo > operadores > comparacion.php. The code editor shows the following PHP code:

```
1 <?php
2
3 //Operadores de Comparación
4
5 //Comparación por Diferencia
6
7 $numero1 = 58;
8
9 $numero2 = 58;
10
11 $resultado = $numero1 != $numero2;
12
13 var_dump($resultado);
14
15 ?>
```

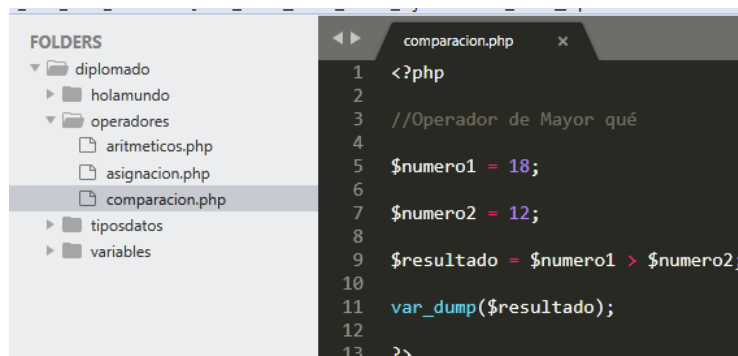


## Operador de Mayor qué

El operador mayor qué, toma ambos valores o variables y compara que el valor o variable de la izquierda sea mayor y sólo mayor que el correspondiente al valor o variable de la derecha. En caso de que lo sea, el resultado será true (verdadero) o en caso contrario false (falso).

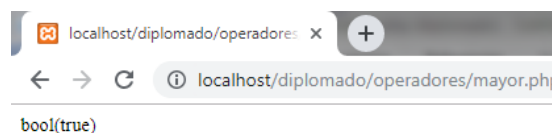
## Haciendo uso del operador mayor qué >

### Caso verdadero

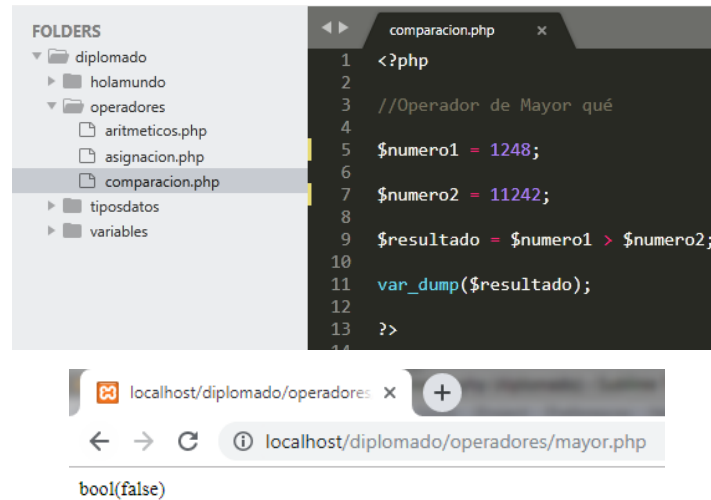


The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a folder structure: diplomado > holamundo > operadores > comparacion.php. The code editor shows the following PHP code:

```
1 <?php
2
3 //Operador de Mayor qué
4
5 $numero1 = 18;
6
7 $numero2 = 12;
8
9 $resultado = $numero1 > $numero2;
10
11 var_dump($resultado);
12
13 ?>
```



## Caso falso



The screenshot shows a code editor with a file named `comparacion.php`. The code is as follows:

```
1 <?php
2
3 //Operador de Mayor qué
4
5 $numero1 = 1248;
6
7 $numero2 = 11242;
8
9 $resultado = $numero1 > $numero2;
10
11 var_dump($resultado);
12
13 ?>
```

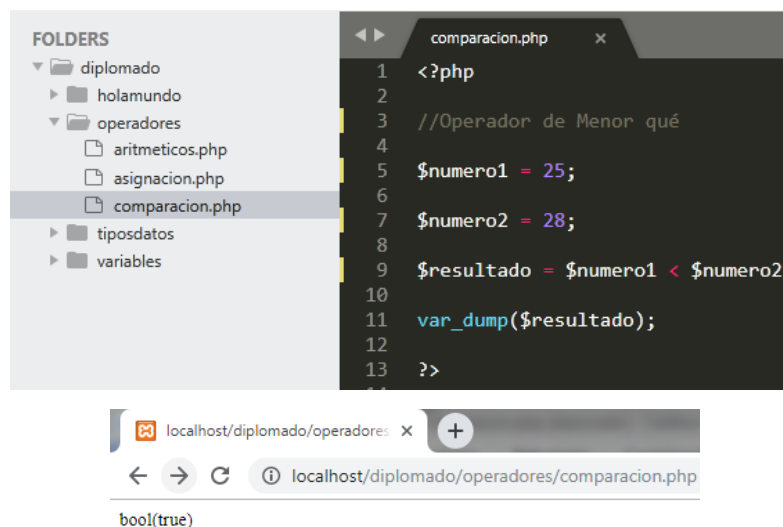
Below the code editor, a browser window is shown with the URL `localhost/diplomado/operadores/mayor.php`. The output displayed is `bool(false)`.

## Operador de Menor qué

El operador menor qué, toma ambos valores o variables y compara que el valor o variable de la izquierda sea menor y sólo menor que el correspondiente al valor o variable de la derecha. En caso de que lo sea, el resultado será true (verdadero) o en caso contrario false (falso).

## Haciendo uso del operador menor qué <

### Caso verdadero



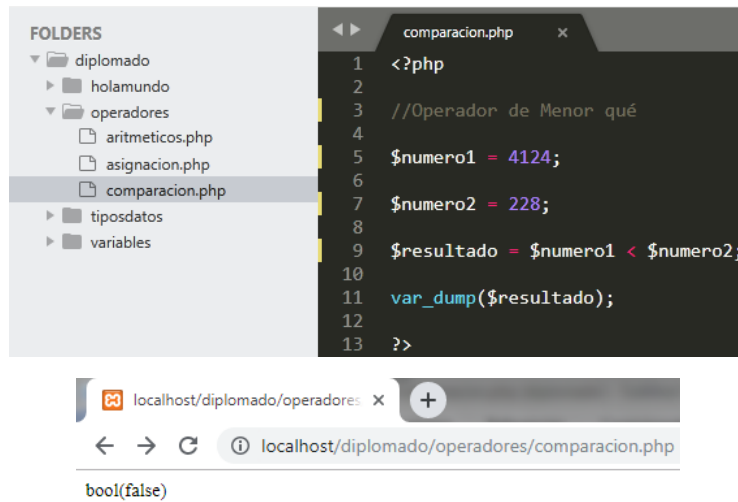
The screenshot shows a code editor with a file named `comparacion.php`. The code is as follows:

```
1 <?php
2
3 //Operador de Menor qué
4
5 $numero1 = 25;
6
7 $numero2 = 28;
8
9 $resultado = $numero1 < $numero2;
10
11 var_dump($resultado);
12
13 ?>
```

Below the code editor, a browser window is shown with the URL `localhost/diplomado/operadores/comparacion.php`. The output displayed is `bool(true)`.



## Caso falso



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a directory structure: diplomado > holamundo > operadores > comparacion.php. The code editor shows the following PHP code:

```
1 <?php
2
3 //Operador de Menor qué
4
5 $numero1 = 4124;
6
7 $numero2 = 228;
8
9 $resultado = $numero1 < $numero2;
10
11 var_dump($resultado);
12
13 ?>
```

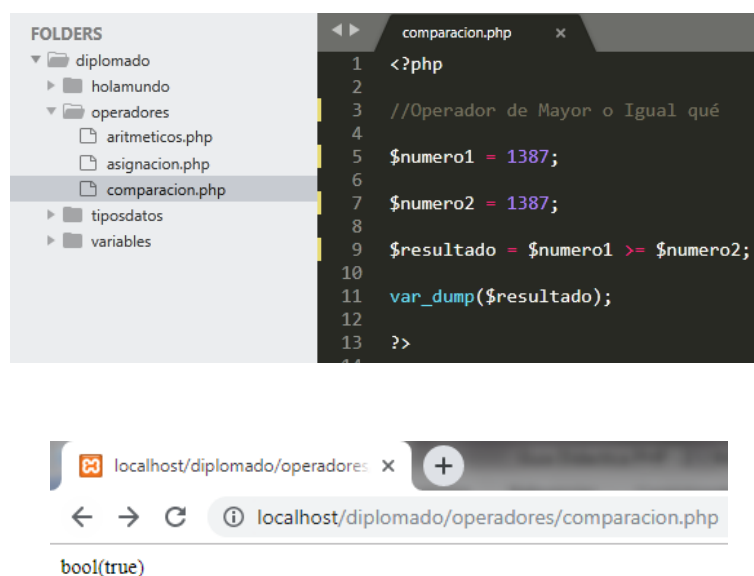
Below the code editor, a browser window is shown with the URL `localhost/diplomado/operadores/comparacion.php`. The output of the script is `bool(false)`.

## Operador de Mayor o Igual qué

El operador mayor o igual qué, toma ambos valores o variables y compara que el valor o variable de la izquierda sea mayor o igual que el correspondiente al valor o variable de la derecha. En caso de que lo sea, el resultado será true (verdadero) o en caso contrario false (falso).

### Haciendo uso del operador mayor o igual qué `>=`

## Caso verdadero

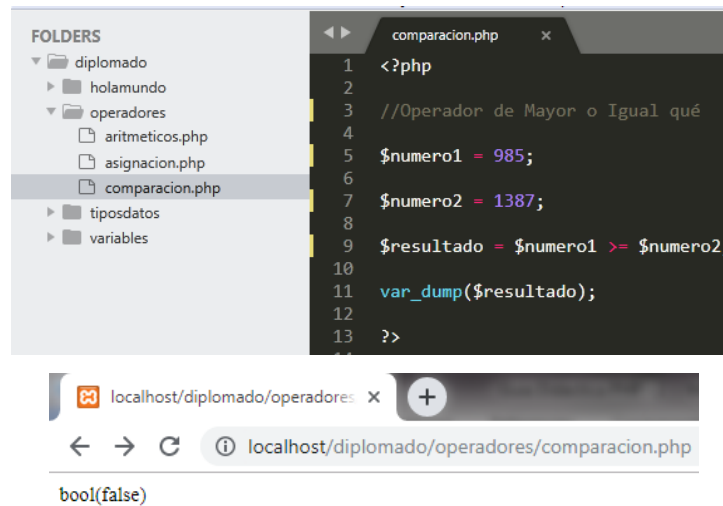


The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a directory structure: diplomado > holamundo > operadores > comparacion.php. The code editor shows the following PHP code:

```
1 <?php
2
3 //Operador de Mayor o Igual qué
4
5 $numero1 = 1387;
6
7 $numero2 = 1387;
8
9 $resultado = $numero1 >= $numero2;
10
11 var_dump($resultado);
12
13 ?>
```

Below the code editor, a browser window is shown with the URL `localhost/diplomado/operadores/comparacion.php`. The output of the script is `bool(true)`.

## Caso falso



The screenshot shows a code editor with a file explorer on the left. The file explorer shows a directory structure: diplomado > holamundo > operadores > comparacion.php. The code editor shows the following PHP code:

```
1 <?php
2
3 //Operador de Mayor o Igual qué
4
5 $numero1 = 985;
6
7 $numero2 = 1387;
8
9 $resultado = $numero1 >= $numero2;
10
11 var_dump($resultado);
12
13 ?>
```

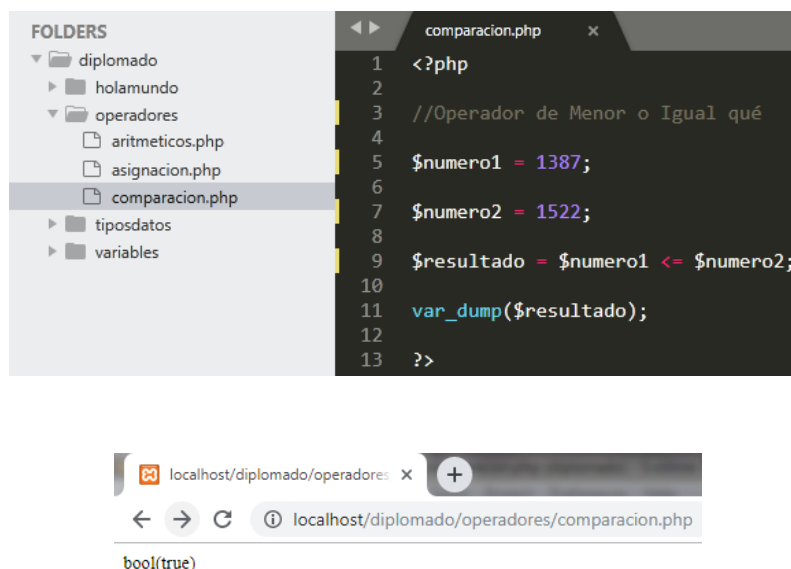
Below the code editor, a browser window is shown with the URL `localhost/diplomado/operadores/comparacion.php`. The output of the script is `bool(false)`.

## Operador de Menor o Igual qué

El operador menor o igual qué, toma ambos valores o variables y compara que el valor o variable de la izquierda sea menor o igual que el correspondiente al valor o variable de la derecha. En caso de que lo sea, el resultado será true (verdadero) o en caso contrario false (falso).

## Haciendo uso del operador menor o igual qué <=

### Caso verdadero

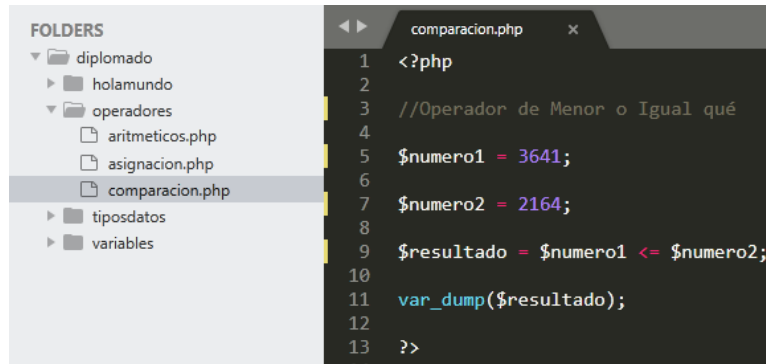


The screenshot shows a code editor with a file explorer on the left. The file explorer shows a directory structure: diplomado > holamundo > operadores > comparacion.php. The code editor shows the following PHP code:

```
1 <?php
2
3 //Operador de Menor o Igual qué
4
5 $numero1 = 1387;
6
7 $numero2 = 1522;
8
9 $resultado = $numero1 <= $numero2;
10
11 var_dump($resultado);
12
13 ?>
```

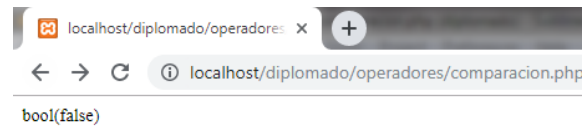
Below the code editor, a browser window is shown with the URL `localhost/diplomado/operadores/comparacion.php`. The output of the script is `bool(true)`.

## Caso falso



```

1 <?php
2
3 //Operador de Menor o Igual qué
4
5 $numero1 = 3641;
6
7 $numero2 = 2164;
8
9 $resultado = $numero1 <= $numero2;
10
11 var_dump($resultado);
12
13 ?>
  
```



## Operadores de Incremento y Decremento

Los operadores de incremento de PHP se utilizan para incrementar el valor de una variable.

Los operadores de disminución de PHP se utilizan para disminuir el valor de una variable.

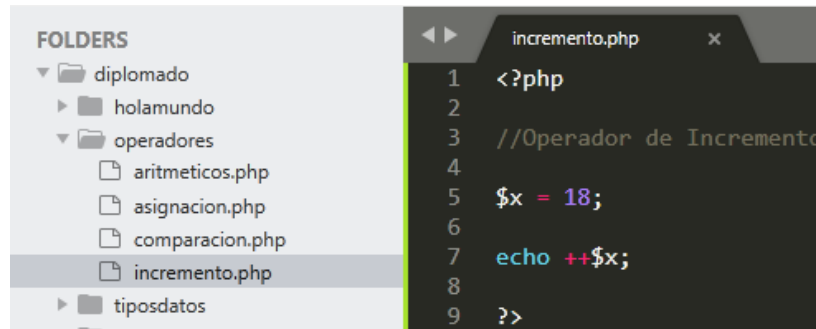
Operación	Nombre	Descripción
++\$x	Pre Incremento	Incrementa \$x en razón de 1 y luego devuelve \$x
\$x++	Pos Incremento	Devuelve \$x y luego incrementa \$x en razón de 1
--\$x	Pre Decremento	Decrementa \$x en razón de 1 y luego devuelve \$x
\$x--	Pos Decremento	Devuelve \$x y luego decrementa \$x en razón de 1

## Operadores de Incremento

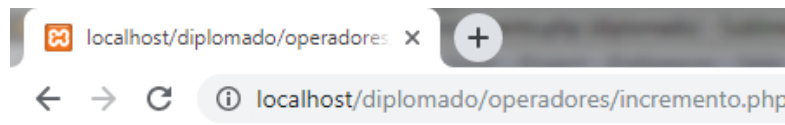
Los operadores de incremento de PHP se utilizan para incrementar el valor de una variable.

### Pre Incremento

Incrementa la variable en razón de 1 y luego retorna la variable.

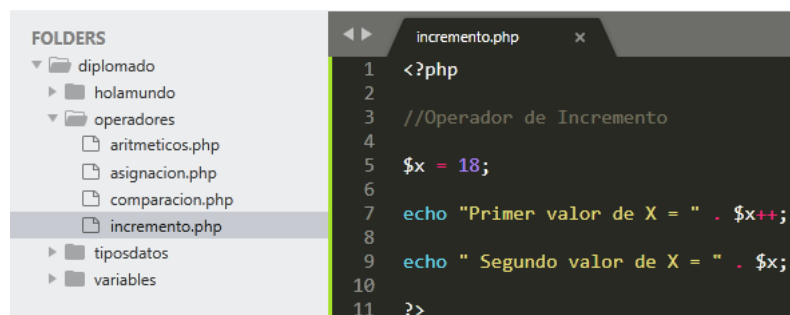


```
1 <?php
2
3 //Operador de Incremento
4
5 $x = 18;
6
7 echo ++$x;
8
9 ?>
```

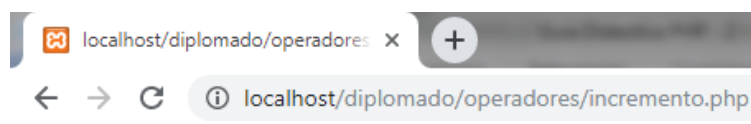


### Pos Incremento

Retorna la variable y luego retorna la variable incrementando a razón de 1.



```
1 <?php
2
3 //Operador de Incremento
4
5 $x = 18;
6
7 echo "Primer valor de X = " . $x++;
8
9 echo " Segundo valor de X = " . $x;
10
11 ?>
```



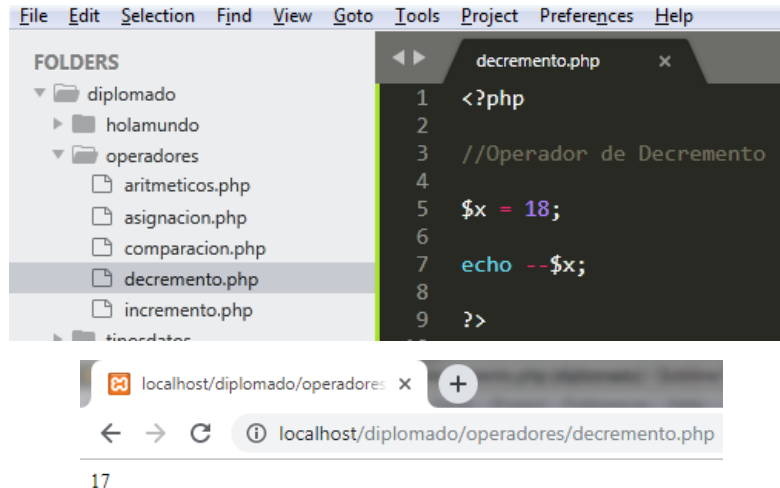
Primer valor de X = 18 Segundo valor de X = 19

## Operadores de Decremento

Los operadores de disminución de PHP se utilizan para disminuir el valor de una variable.

### Pre Decremento

Decrementa la variable en razón de 1 y luego retorna la variable.



```
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
  diplomado
    holamundo
    operadores
      aritmeticos.php
      asignacion.php
      comparacion.php
      decremento.php
      incremento.php
      tiposdatos
  variables
decremento.php
1 <?php
2
3 //Operador de Decremento
4
5 $x = 18;
6
7 echo --$x;
8
9 ?>
```

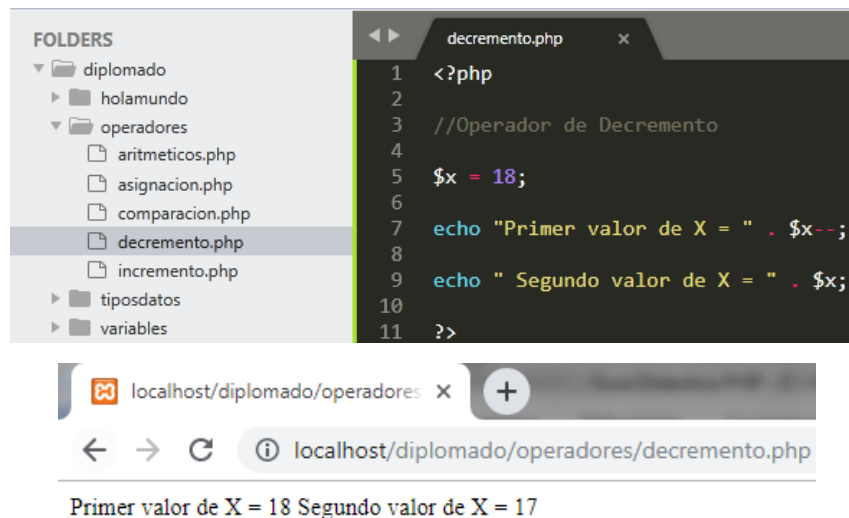
localhost/diplomado/operadores: x +

localhost/diplomado/operadores/decremento.php

17

### Pos Decremento

Retorna la variable y luego retorna la variable decrementando a razón de 1.



```
FOLDERS
  diplomado
    holamundo
    operadores
      aritmeticos.php
      asignacion.php
      comparacion.php
      decremento.php
      incremento.php
      tiposdatos
  variables
decremento.php
1 <?php
2
3 //Operador de Decremento
4
5 $x = 18;
6
7 echo "Primer valor de X = " . $x--;
8
9 echo " Segundo valor de X = " . $x;
10
11 ?>
```

localhost/diplomado/operadores: x +

localhost/diplomado/operadores/decremento.php

Primer valor de X = 18 Segundo valor de X = 17

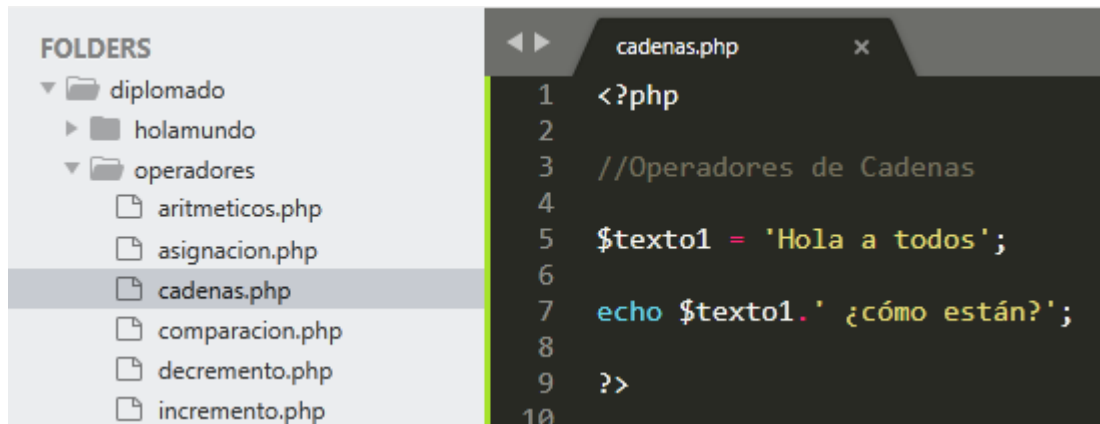
## Operadores con Cadenas

PHP tiene dos operadores que están especialmente diseñados para cadenas.

Operación	Nombre	Ejemplo
.	Concatenación	\$texto1 . "texto"
.=	Concatenación y Asignación	\$texto1 .= \$texto2

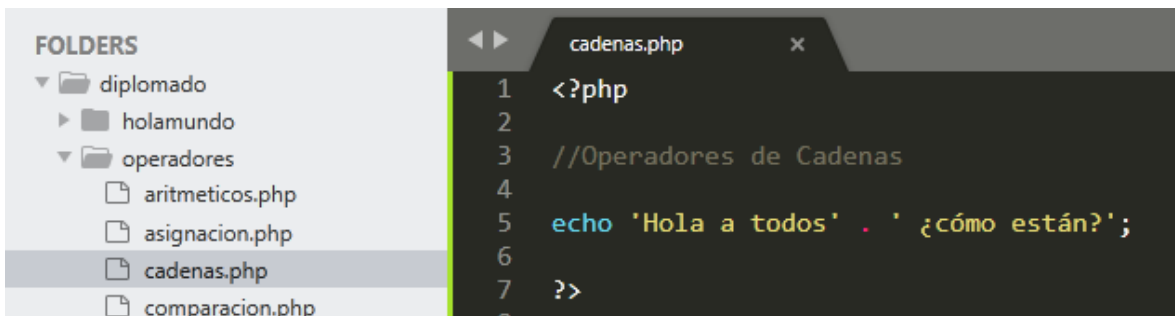
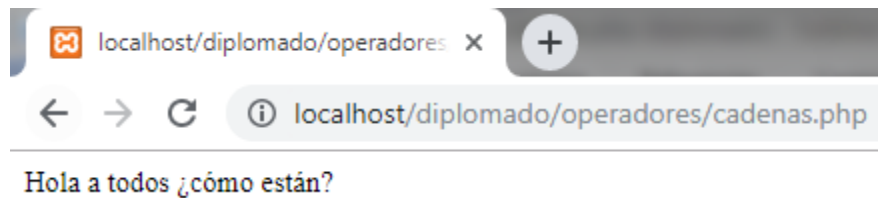
### Operador de Concatenación

El operador de concatenación "." permite unir dos cadenas de texto.



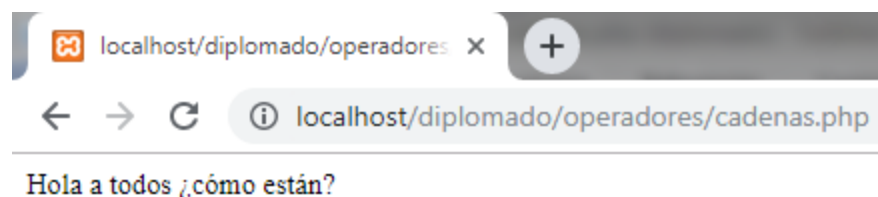
The screenshot shows a code editor with a file named 'cadenas.php'. The code is as follows:

```
1 <?php
2
3 //Operadores de Cadenas
4
5 $texto1 = 'Hola a todos';
6
7 echo $texto1.' ¿cómo están?';
8
9 ?>
```



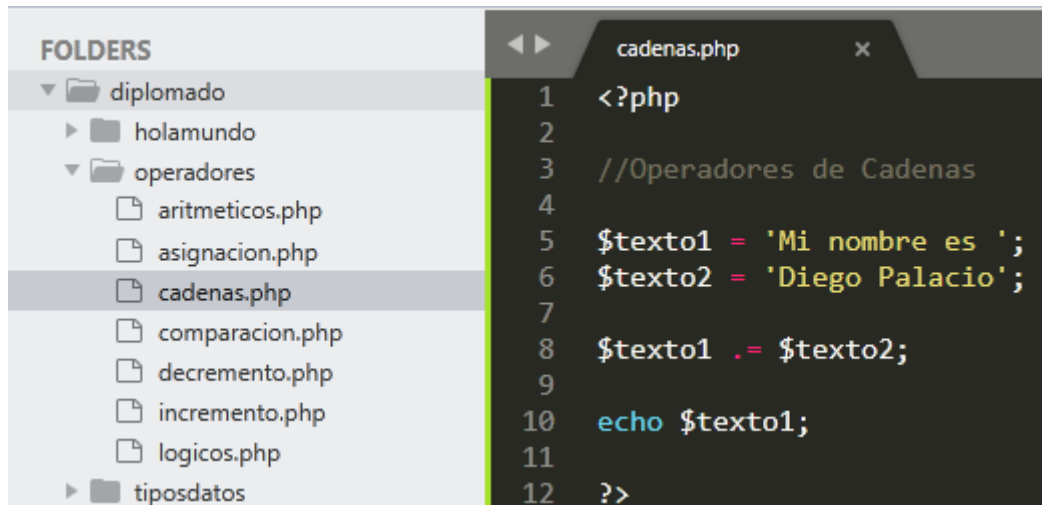
The screenshot shows a code editor with a file named 'cadenas.php'. The code is as follows:

```
1 <?php
2
3 //Operadores de Cadenas
4
5 echo 'Hola a todos' . ' ¿cómo están?';
6
7 ?>
```



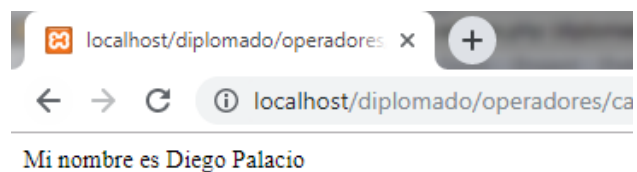
## Operador de Concatenación y Asignación

El operador de concatenación y asignación “.=” permite unir dos cadenas de texto, en la declaración o re declaración de una variable.



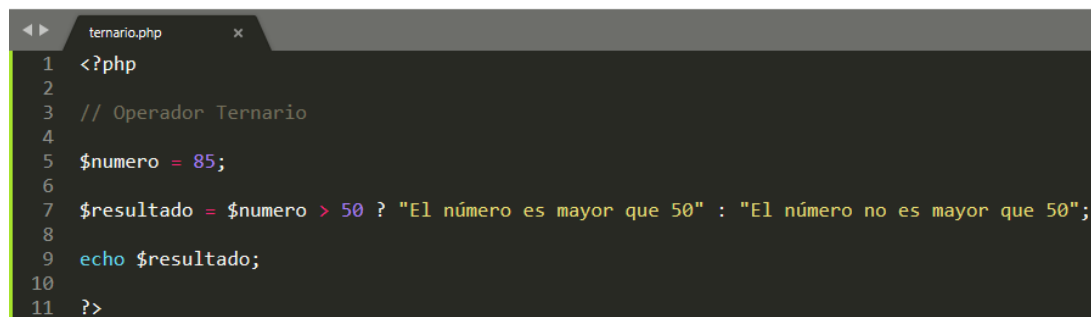
```
FOLDERS
└─ diplomado
   └─ operadores
      └─ aritmeticos.php
      └─ asignacion.php
      └─ cadenas.php
      └─ comparacion.php
      └─ decremento.php
      └─ incremento.php
      └─ logicos.php
      └─ tiposdatos

cadenas.php
1 <?php
2
3 //Operadores de Cadenas
4
5 $texto1 = 'Mi nombre es ';
6 $texto2 = 'Diego Palacio';
7
8 $texto1 .= $texto2;
9
10 echo $texto1;
11
12 ?>
```

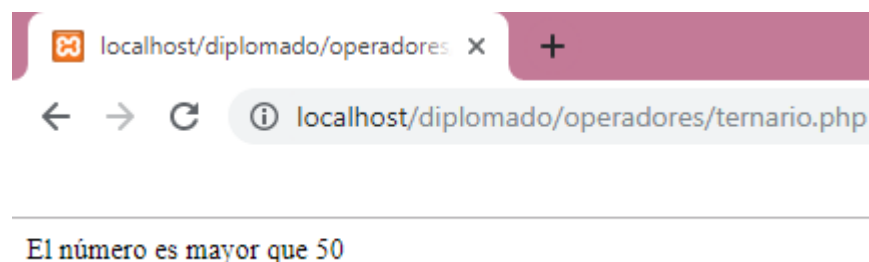


## Operador Ternario

El operador ternario permite establecer valores según una condición.



```
ternario.php
1 <?php
2
3 // Operador Ternario
4
5 $numero = 85;
6
7 $resultado = $numero > 50 ? "El número es mayor que 50" : "El número no es mayor que 50";
8
9 echo $resultado;
10
11 ?>
```



```
ternario.php x
1 <?php
2
3 // Operador Ternario
4
5 $resultado = "Evaluación de condición" ? "Resultado Verdadero" : "Resultado Falso";
6
7 echo $resultado;
8
9 ?>
```

La forma en la que opera el operador ternario es la siguiente, este, cuenta con una condición / evaluación, esta arroja un resultado al igual que los operadores pasados (verdadero o falso), a partir del resultado se define el valor que tomará la variable "resultado"; hay algo que se debe tener muy en cuenta y es que, que la condición / evaluación arroja un resultado verdadero o falso, no significa que la variable tomará ese valor, los resultados se pueden definir y parametrizar, desde un simple número, hasta una cadena de texto puede ser y claramente, sí el resultado que se espera es verdadero o falso, se puede parametrizar, aquí algunos ejemplos.

```
ternario.php x
1 <?php
2
3 // Operador Ternario
4
5 $resultado = true ? "Verdadero" : "Falso";
6
7 echo $resultado;
8
9 ?>
```

localhost/diplomado/operadores x +

← → ↻ ⓘ localhost/diplomado/operadores/ternario.php

Verdadero



```
ternario.php x
1 <?php
2
3 // Operador Ternario
4
5 $nombrePersona = "Diego";
6
7 $resultado = $nombrePersona == "Juan" ? "Es Juan" : "No es Juan";
8
9 echo $resultado;
10
11 ?>
```

localhost/diplomado/operadores x +

← → ↻ ⓘ localhost/diplomado/operadores/ternario.php

---

No es Juan

```
ternario.php x
1 <?php
2
3 // Operador Ternario
4
5 $n1 = 14;
6 $n2 = 9;
7
8 $resultado = $n1 > $n2 ? "Multiplicación de ambos números " . $n1*$n2 : "Error";
9
10 echo $resultado;
11
12 ?>
```

localhost/diplomado/operadores x +

← → ↻ ⓘ localhost/diplomado/operadores/ternario.php

---

Multiplicación de ambos números 126

**Nota:** como se detalla en estos ejemplos, el resultado de la condición / evaluación es claramente diferente a los que se observa en los operadores

vistos anteriormente, donde sólo existían dos posibles resultados, verdadero o falso. El operador ternario permite jugar un poco con esto, los resultados a partir de él, son configurables y se pueden obtener otros resultados gracias a este.

### Ejercicio

¿Deseas profundizar en la temática de Operadores? Entonces te sugiero realizar los siguientes ejercicios que pondrán a prueba los conocimientos adquiridos.

#### **(MÓDULO 1 – EJERCICIOS CON OPERADORES)**

¡Inténtalo! 👍



**¡Felicidades! 👍 Recuerda que si tienes una duda o dificultad puedes escribirme: [diegovalencia@politecnicodecolombia.edu.co](mailto:diegovalencia@politecnicodecolombia.edu.co).**