



Sentencias PHP

PHP al igual que la mayoría de lenguajes de programación necesita de herramientas que permitan tomar decisiones lógicas al servidor.

1. If...Else
2. If...Elseif...Else
3. Switch

Estas son todas las posibilidades condicionales que nos permite este lenguaje de programación.

If...Else[\[editar\]](#)

Estas instrucciones son las mas usadas de todos los lenguajes de programación, ya que son las mas básicas. IF (Español: SI) ELSE (Español: SI NO).

Estructura:

```
if(expresión)
{
    ... Sentencias;
}
else
{
    ... Sentencias;
}
```

De momento, en ese cachito de código aunque sin nada tangible, estarías diciendo que si la expresión se cumple se realizan sentencias, y en este caso si no se cumpliese la expresión se evaluaría las sentencias que están dentro de **else**{...}.

Un ejemplo para probar en casa:

```
<?php
$galletas = true;
if($galletas)
{
    echo 'Hay galletas';
}
else
{
    echo 'No hay galletas';
}
?>
```

En este ejemplo, estamos dando un valor *\$galletas* y comprobamos si es cierto o no, introduciendolo en la expresión del condicional. Tal cual está, este ejemplo



produciría el resultado 'Hay galletas', aunque si cambiasemos `$galletas = true`; por `$galletas = false` daría como resultado la segunda sentencia.

If...Elseif...Else[[editar](#)]

La estructura de estas instrucciones viene a ser muy similar que la anterior, y de hecho el ejemplo anterior es válido igualmente para este.

Estructura:

```
<?php
if(expresión)
{
    ... Sentencias;
}
elseif(expresión)
{
    ... Sentencias;
}
else
{
    ... Sentencias;
}
?>
```

Esta es la sintaxis con la que debe ser usada **elseif**. Un **elseif** se ejecutará, siempre y cuando, no haya alguna respuesta verdadera en el **if** o **elseif** anterior; luego de que se llega al final de los **elseif**, y ninguno ha sido verdadero, se ejecutará lo que está dentro de el **else** final.

Pueden haber varios **elseif**, pero estos estan siempre dentro de un **if** y un **else**(sólo uno) final.

Otro ejemplo, para dejar en claro el funcionamiento de **elseif**:

```
<?php
$test = 33;
if ($test > 40) {
    echo "Sí, $test es mayor que 40.";
}
elseif ($test > 35)
    echo "Sí, $test es mayor que 35.";
}
elseif ($test > 30) {
    echo "Sí, $test es mayor que 30.";
}
else {
    echo "No, $test es menor que 40, 35 y 30.";
}
```



?>

En este caso la respuesta sería:

Sí, 33 es mayor que 30.

Si el valor de **\$test** se cambia a 22, la respuesta sería:

No, 22 es menor que 40, 35 y 30.

Switch[[editar](#)]

Las instrucciones **If...Elseif...Else** permiten seleccionar entre más de dos posibles condiciones que puedan ser evaluadas como true o false, pero en ocasiones nos encontraremos con casos en que queremos evaluar condiciones con una estructura como esta:

```
if($variable==Valor1)
{
    ... Sentencias;
}
elseif($variable==Valor2)
{
    ... Sentencias;
}
elseif($variable==Valor3)
{
    ... Sentencias;
}
else
{
    ... Sentencias;
}
```

En las que esperamos realizar una acción determinada según el valor tomado por una variable. Para estos casos la mayoría de los lenguajes de programación incluyen una instrucción que permite "seleccionar" entre los posibles valores de una variable: la instrucción **Switch** (Español: SELECCIONAR).

Estructura:

```
switch($variable)
{
    case Valor1:
        ... Sentencias;
        break;

    case Valor2:
```



```
... Sentencias;
break;

case Valor3:
... Sentencias;
break;

.
.
.

default:
... Sentencias;
}
```

En esta estructura, \$variable es la variable que será comparada sucesivamente con cada una de las instrucciones **case** hasta hallar la que corresponda al valor de \$variable, una vez que la encuentre se ejecutarán sucesivamente todas las instrucciones pertenecientes al **switch** hasta hallar la siguiente instrucción **break**; esta última hace un salto pasando el control del programa a la instrucción inmediata siguiente fuera del **switch**.

Si ninguna de las instrucciones **case** corresponde al valor de \$variable se ejecutarán solo las instrucciones siguientes al **default**. Al igual que **else**, la instrucción **default** es opcional.

Veamos un ejemplo:

| Usando if...elseif...else | Usando switch |
|---|--|
| <pre><?php \$variable=2; if(\$variable==1) { echo '\$Variable es igual a 1.'; } elseif(\$variable==2) { echo '\$Variable es igual a 2.'; } elseif(\$variable==3) { echo '\$Variable es igual a 3.'; } else { }</pre> | <pre><?php \$variable=2; switch(\$variable) { case 1: echo '\$Variable es igual a 1.'; break; case 2: echo '\$Variable es igual a 2.'; break; case 3: echo '\$Variable es igual a 3.'; break; }</pre> |



```
echo '$Variable no es igual ' .  
    'a 1, 2 o 3.';  
}  
?>
```

```
default:  
echo '$Variable no es igual ' .  
    'a 1, 2 o 3.';  
}  
?>
```

El resultado de ambas formas mostradas será:

\$Variable es igual a 2.

Si por el contrario cambiamos \$Variable=2 por \$Variable=5 al inicio de código el resultado sería:

\$Variable no es igual a 1, 2 o 3.

Una diferencia importante respecto a las instrucciones **if...elseif...else** es que se requiere de la instrucción **break** para salir del **switch** al terminar las instrucciones correspondientes a un **case**. Esto es así debido a que solo podemos especificar un valor en cada **case**. Veamos un ejemplo:

| Usando if...elseif...else | Usando switch |
|--|--|
| <pre><?php \$variable=2; if(\$variable==1) { echo '\$Variable es igual a 1.'; } elseif(\$variable==2 OR \$variable==3) { echo '\$Variable es igual a 2 o 3.'; }else { echo '\$Variable no es igual ' . 'a 1, 2 o 3.'; } ?></pre> | <pre><?php \$variable=2; switch(\$variable) { case 1: //Se ejecuta si \$variable vale 1 echo '\$Variable es igual a 1.'; break; case 2: case 3: //Se ejecuta si \$variable vale 2 o 3 echo '\$Variable es igual a ' . '2 o 3.'; break; default: //Se ejecuta en cualquier otro caso</pre> |



```
echo '$Variable no es igual ' .  
    'a 1, 2 o 3.';  
}  
?>
```

En cualquiera de las dos formas mostradas, la salida obtenida sería:

```
$Variable es igual a 2 o 3.
```

En este caso, al evaluarse la segunda instrucción **case** no se encuentra ningún **break** por lo que siguen ejecutandose las instrucciones siguientes al tercer **case**, aun cuando \$Variable es diferente de 3. Más aún, si borramos el **break** del tercer **case** también se ejecutarán las instrucciones siguientes a **default**, y la salida sería entonces:

```
$Variable es igual a 2 o 3.  
$Variable no es igual a 1, 2 o 3.
```