Operadors PHP

- **Operadors de comparació (valors i tipus de dades)**

  Els operadors de comparació són fonamentals per crear condicions en les estructures de control de flux. Permeten comparar valors i prendre decisions en funció dels resultats de les comparacions.

  - == Igualtat de valor (ex. 5 == '5' retorna true perquè compara només el valor).
  - === Igualtat estricta (valor i tipus) (ex. 5 === '5' retorna false perquè el tipus no és igual).
  - != Diferència de valor (ex. 5 != '6' retorna true).
  - !== Diferència estricta (ex. 5 !== '5' retorna true perquè compara valor i tipus).
  - <, >, <=, >= Comparació numèrica i alfabètica.
  - <=> Operador "nave espacial" o "de tres vies" (retorna -1, 0 o 1 segons si el valor de l'esquerra és menor, igual o major que el de la dreta).

Operadors PHP

- **Operadors lògics (Combinació de Condicions)**

  Els operadors lògics permeten combinar múltiples condicions en una sola expressió, fent possible definir condicions complexes. Aquests operadors són útils en estructures if, while, o for quan es vol verificar més d'una condició alhora.

  - &&: I lògic (retorna true si ambdues condicions són true).
  - ||: O lògic (retorna true si almenys una condició és true).
  - !: Negació lògica (inverteix el valor lògic).
  - and i or: Alternatives a && i || amb menor prioritat en la jerarquia de PHP.

Operadors PHP

- **Operadors ternaris**

  Els operadors ternaris ofereixen una manera compacta i ràpida d'executar operacions condicionals en una sola línia. Es poden veure com una alternativa curta a les estructures if-else, útil quan hi ha només una condició simple i dos resultats possibles.

  Sintaxi

  $variable = (condició) ? valor_si_cert : valor_si_fals;

  Exemple:

  ```
  $edat = 20;
  $missatge = ($edat >= 18) ? "Accés permès" : "Accés denegat";
  echo $missatge; // Resultat: "Accés permès"
  ```

Estructures de control de flux

Sentències de presa de decisions:

- if, else, elseif
- switch

Bucles:

- for
- while
- do… while
- foreach

Sentències de presa de decisions

Sentències IF, ELSE, ELSEIF

Utilitzem **if** per comprovar una condició inicial, **elseif** per afegir condicions addicionals, i **else** per cobrir qualsevol altre cas no contemplat.

```
if (condició) {

    // Codi a executar si la condició és certa

} elseif (altra_condició) {

    // Codi a executar si la segona condició és certa

} else {

    // Codi a executar si cap condició anterior és certa

}
```

Sentències de presa de decisions

Sentència SWITCH

És més clar quan cal comparar una variable amb diversos valors possibles. És més eficient per a condicions simples on la variable només pot tenir un o uns valors concrets.

```
switch (variable) {

    case valor1:
        // Codi a executar si variable és igual a valor1
        break;
    case valor2:
        // Codi a executar si variable és igual a valor2
        break;
    default: // és opcional
        // Codi a executar si cap cas és complert
        break;
}
```

Sentències de presa de decisions

Sentència SWITCH (Exemple)

```
$tipusUsuari = "premium";

switch ($tipusUsuari) {
    case "administrador":
        echo "Benvingut administrador! Accés complet.<br>";
        break;
    case "premium":
        echo "Benvingut usuari premium! Accés especial.<br>";
        break;
    default:
        echo "Tipus d'usuari desconegut.<br>";
        break;
}
```

Sentències de control de bucles

Sentència FOR

s'utilitza quan sabem el nombre exacte de vegades que volem executar el codi.

```
for (inicialització; condició; increment) {
        instrucció;         // Codi a executar mentre la condició sigui certa
}
```

```
for (inicialització; condició; increment)
        instrucció;         // Codi a executar mentre la condició sigui certa
```

```
for (inicialització; condició; increment):
        instrucció;         // Codi a executar mentre la condició sigui certa
endfor;
```

Sentències de control de bucles

Sentència FOR (Exemple)

```
$num = 5;

echo "Taula de multiplicar del $num:<br>";
for ($i = 1; $i <= 10; $i++) {
        echo "$num x $i = " . ($num * $i) . "<br>";
}


for ($i = 1; $i <= 10; $i++)
echo "$num x $i = " . ($num * $i) . "<br>";


for ($i = 1; $i <= 10; $i++):
        echo "$num x $i = " . ($num * $i) . "<br>";
endfor;
```

Sentències de control de bucles

Sentència WHILE

És útil quan no sabem quantes vegades es complirà la condició i volem que el codi s'executi mentre la condició sigui certa.

```
while (condició) {
    // Codi a executar mentre la condició sigui certa
}
```

Exemple:

```
$inici = 10;
while ($inici > 0) {
    echo "$inici<br>";
    $inici--;
}
echo "Temps esgotat!";
```

Sentències de control de bucles

Sentència DO… WHILE

S'assegura que el codi s'executi almenys una vegada, independentment de si la condició és certa.

```
do {

    // Codi a executar almenys una vegada

} while (condició);
```

Sentències de control de bucles

Sentència FOREACH

És ideal per recórrer arrays i treballar amb cada element. La relació és tanta que, si utilitzem un tipus de dada que no sigui un array, ens donarà error.

```
foreach ($array as $valor) {

    // Codi a executar per cada element de l'array

}

foreach ($array as $clau => $valor) {

    // Codi a executar per cada element de l'array

}
```

Sentències de control de bucles

Sentència FOREACH (Exemple)

```
$notes = [4, 7, 3, 6, 5, 8, 9, 2, 7, 10];
$aprovats = 0;
$suspensos = 0;

foreach ($notes as $nota) {
    if ($nota >= 5) {
        $aprovats++;
    } else {
        $suspensos++;   }
}
echo "Nombre d'aprovats: $aprovats<br>";
echo "Nombre de suspensos: $suspensos";
```