

Entorns de Desenvolupament



7. Bucles

7. Bucles

Les **estructures repetitives** (bucles) són aquelles que reiteren una o un grup d'instruccions "n" vegades i depenen d'una variable de control del cicle. És a dir, executen una o diverses instruccions un nombre de vegades definit. Les instruccions bàsiques que permeten construir aquest tipus d'estructures són **while**, **do ... while** i **for**.

La instrucció while

La instrucció **while** ("**mentre ...**") executa una porció de programa mentre es complisca una certa condició.

Mentre la **condició** siga **veritat**, **s'executen** les **instruccions contingudes** en el **while**. Quan deixa de complir-se la condició, s'ix del cicle i es continua executant la resta del programa.

Si la **condició no es compleix ni** la **primera vegada** que es comprova, les **línies** a l'interior del **while no s'executaran mai**.

L'estructura general o sintaxi habitual d'un bucle **while** és la següent:

```
<?php
    while (condició) {
        ...
        Sentències;
        ...
    }
?>
```

En aquest fragment de codi estaríem indicant que mentre es complisca la condició es realitzen les sentències, i en aquest cas si no es complira la condició se seguiria executant el programa o mostrant els continguts de la pàgina web just després del tancament de la instrucció **while** (que queda determinat pel símbol de tancament **}**).

Exemple 01:

```
<?php
    $comptador = 0; //inicialització variable control
    while ($comptador < 10 ) { //comprovació condició
        echo "Comptador while: $comptador<br/>";
        $comptador++; //modificació variable control
    }
?>
```

En aquest exemple, les instruccions dins el **while** s'executarien 10 vegades, mostrant els números de 0 a 9 en línies consecutives. **Cal tenir especial cura a no crear estructures repetitives que s'executen infinitament (bucles infinits).** En l'exemple anterior, **si no incrementem el valor de la variable \$comptador**, la condició del **while** es compliria sempre i tindríem un problema perquè hauríem creat un bucle infinit, i el programa 'encallat' en aquest **while**.

```
<?php
    $comptador = 0; //inicialització variable control
    while ($comptador < 10 ) { //comprovació condició
        echo "Comptador while: $comptador<br/>";
        //bucle infinit, no incrementem $comptador
    }
?>
```

La instrucció do ... while

L'estructura repetitiva **do ... while** ("fer ... mentre") és molt **similar** a l'estructura **while**, excepte que l'expressió condicional és verificada al final de cada iteració en lloc de al principi. La **diferència** principal amb els bucles **while** és que està garantit que **s'executen** les instruccions que contenen, **almenys una vegada** perquè la verificació de si s'ha de repetir el procés es realitza al final de la repetició de l'estructura.

L'estructura general o sintaxi habitual d'un bucle **do - while** és la següent:

```
<?php
    do {
        ...
        Sentències
        ...
    } while (condició);
?>
```

En aquest fragment de codi estaríem indicant que **s'execute** una **primera vegada** "sí o sí" les **sentències**, i que després s'avalua la condició per a comprovar si s'han d'executar novament les sentències. Com veiem, encara que **no** es **complisca** la **condició** les sentències **s'executarien almenys una primera vegada**.

Exemple 02:

```
<?php
$comptador = 0; //inicialització variable control
do {
    echo "Comptador do-while: $comptador<br/>";
    $comptador ++; //modificació variable control
} while ($comptador < 10); //comprovació condició
?>
```

La instrucció for

L'estructura repetitiva **for** ("per a ...") s'utilitza generalment quan tenim ben **determinada** la **quantitat** de **repeticions** a realitzar. Es diferencia de les anteriors en què s'ha d'**incloure** en la pròpia **instrucció** una **variable** de **control**, la qual s'**incrementa** o **decrementa** de forma **automàtica**. L'estructura d'aquestes instruccions ve a ser molt similar a l'anterior.

L'estructura general o sintaxi habitual d'un bucle **for** és la següent:

```
<?php
    for ($variable = valor_inicial; condició; increment) {
        ...
        Sentències;
        ...
    }
?>
```

En aquest fragment de codi estaríem indicant que mentre es complisca la condició es realitzen les sentències, i en aquest cas si no es complira la condició se seguiria executant el programa just després del tancament de la instrucció **for (}**). Tot això és **igual** a la **instrucció while** anteriorment citada, **excepte** que en la pròpia **declaració** de **l'estructura** es troba la **variable** de **control**, que en cada repetició s'incrementa automàticament.

Aquesta **instrucció** consta, per tant, de tres **paràmetres**:

- **Inicialització** de la **variable** que s'utilitzarà en la condició.
- **Condició** que s'ha de complir per a continuar executant el cicle.
- **Modificació** de la **variable** utilitzada en la condició.

Després de la inicialització i la condició hem d'**introduir un punt i coma**, com si d'instruccions habituals foren, però després de l'última, no s'han de ficar.

Si només volem executar una instrucció, les claus no són necessàries, però és recomanable introduir-les per fer més entenedor el codi del programa.

Exemple 3:

```
<?php
    for($comptador=0; //inicialització variable control
        $comptador<10; //comprovació condició
        $comptador++) { //modificació variable control
        echo "Comptador for: $comptador<br/>";
    }
?>
```

Aquest exemple mostra a l'eixida, la mateixa informació que mostrava la instrucció **while**. Com podem observar, el valor inicial i l'increment de la variable **\$comptador**, van en la mateixa instrucció.

Modificar el pas (step) de la variable comptadora

És possible fer que l'**increment** o **decrement** de la **variable** de control siga **diferent d'1**. Simplement s'ha de canviar la instrucció en la qual es modifica aquesta variable després de la comprovació de la condició.

Exemple 04:

```
<?php
    for($i=0; $i<50 ; $i+=5) {
        echo "Variable i: $i <br/>" ;
    }
?>
```

En aquest cas a la variable comptador l'hem anomenada **i**. És habitual utilitzar noms com **i**, **j**, **k** per les variables comptadores en bucles.

Usant la instrucció **+=** aconseguim que la variable comptadora no vaja d'un en un, sinó que el pas (step) de la variable siga la quantitat que nosaltres desitgem. En aquest cas, hem utilitzat un increment de 5 unitats en cada repetició del bucle. També podem aconseguir el mateix efecte però amb decrements usant **-=**.