

Structuri de control

Aceste structuri permit definirea unei conditii si apoi executarea unor operatii functie de raspuns.

Structurile de control sunt:

if	while	require
else	for	break
elseif	foreach	continue
switch	include	return

if este instructiunea conditionala cea mai folosita avand urmatoarea sintaxa:

if (conditie)

```
{  
    instructiuni executate daca este indeplinita conditia ;  
}
```

Conditia este incadrata de paranteze rotunde iar instructiunea care se executa in cazul in care conditia este indeplinita intre acolade. Neindeplinirea conditiei face sa se execute ce urmeaza dupa inchiderea acoladei.

Exemplu: se compara doua variabile si se afiseaza un mesaj

```
<?php  
$a=4;  
$b=3;  
echo "$a=4<br>";  
echo "$b=3<br>";  
if ($a>$b)  
{  
    echo "$a>$b";  
}  
?>
```

else este alta instructiunea conditionala care indica ce se executa in cazul in care conditia mentionata in instructiunea **if** nu este indeplinita si are sintaxa:

if (conditie)

```
{  
    instructiuni executate daca este indeplinita conditia;  
}  
else  
{  
    instructiuni executate daca nu este indeplinita conditia;  
}
```

Exemplu: se compara doua variabile si se afiseaza un mesaj

```
<?php  
$a=2;  
$b=9;  
echo "$a=2<br>";  
echo "$b=9<br>";  
if ($a>$b)
```

```

    {
        echo "$a>$b";
    }
    else
    {
        echo "$a<$b";
    }
?>

```

elseif este o combinatie între **if** și **else**. În cazul în care condiția **if** nu este îndeplinită se introduce **elseif** care testează încă o condiție. Dacă nu este îndeplinită a doua condiție se execută declarația introdusă prin **else**.

```

if (conditia1)
{
    instructiuni executate daca este indeplinita conditia 1;
}
elseif (conditia2)
{
    instructiuni executate daca este indeplinita conditia 2;
}
else
{
    instructiuni executate daca nu este indeplinita conditia 2;
}

```

Exemplu:

```

<?php
$a=19;
$b=23;
if ($a<$b)
{
    echo "$a<$b";
}
elseif ($a==$b)
{
    echo "$a==$b";
}
else
{
    echo "$a>$b";
}
?>

```

switch este asemănătoare funcției **if** dar condiția are mai mult de două valori.

Exemplu: transforma numărul zilei în numele ei

```

<?php
$ziua=2;
echo $ziua;

```

```
echo "<br>";
switch ($ziua)
{
case 1:
    echo "Luni";
    break;
case 2:
    echo "Marti";
    break;
case 3:
    echo "Miercuri";
    break;
case 4:
    echo "Joi";
    break;
case 5:
    echo "Vineri";
    break;
case 6:
    echo "Sambata";
    break;
case 7:
    echo "Duminica";
    break;
default:
    echo "Nu este zi a saptamanii";
}
?>
```

while este o instructiune de tip bucla. Atat timp cat conditia este adevarata se repeta bucla.

Exemplu:

```
<?php
$numar=1;
while ($numar<=7)
{
    echo $numar."<br>";
    $numar++;
}
?>
```

for este o instructiune de tip bucla.

```
for (expresia1; expresia2; expresia3)
{
    instructiuni de executat;
}
```

expresia1 este evaluata la inceputul buclei

expresia2 se verifica la inceputul fiecarei iteratii

expresia3 se executa la sfarsitul fiecarei iteratii

construcția **for** poate funcționa și fără una sau toate aceste expresii

Exemplu: se inițializează variabila *i* cu valoarea 1, se verifică dacă este mai mică sau egală cu 10 iar apoi se incrementează. La fiecare iteratie se tipărește variabila *i*.

```
<?php
for ($i=1; $i<=10; $i++)
{
    echo $i."<br>";
}
?>
```

foreach este o construcție care funcționează doar cu matrice generând erori când sunt folosite variabile cu tipuri de date diferite sau variabile neinițializate. Sintaxa folosită este:

```
foreach ($matrice as $cheie=>$valoare)
{
    instructiuni de executat;
}
```

Exemplu:

```
<?php
$personal = array("Florin" => director, "Catalin" => inginer, "Mihai"
=> economist, "Sorin" => sofer, "Diana" => secretara );

foreach ($personal as $nume => $meserie)
{
    echo "$nume => $meserie<br>";
}
?>
```

include și **require** sunt două funcții asemănătoare folosite pentru includerea în paginile php a unor fișiere externe. Diferența între cele două funcții constă în faptul că dacă **include** da greș scriptul generează o avertizare dar funcționează în continuare în timp ce la **require** se termină executarea scriptului.

```
include "fisier1.php";
require "fisier2.html";
```

break oprește forțat execuția structurilor *for*, *foreach*, *while*, *do..while* sau *switch*. **break** acceptă opțional un argument numeric care indică numărul de structuri imbricate a căror funcționare este oprită.

Exemplu: la *i=6* se oprește execuția buclei

```
<?php
for ($i=0; ;$i++)
{
    if ($i>6)
    {
        break;
    }
}
```

```
    }  
    echo $i."<br>";  
}  
?>
```

continue sare peste restul din iteratia buclei curente si continua executia la inceputul iteratiei urmatoare. **continue** accepta optional un argument numeric care indica numarul de bucle care vor fi sarite pana la sfarsit.

Exemplu: sare executia peste $i==5$

```
<?php  
    for ($i=0;$i<8;$i++)  
    {  
        if ($i==5) continue;  
        echo $i."<br>";  
    }  
?>
```

return

Daca este chemata din interiorul unei functii declaratia `return()` opreste imediat executia functiei curente si furnizeaza argumentul ca valoare a functiei. Daca este chemata in scop global executia scriptului curent se opreste.