

## Ciklet në PHP

Ciklet përdoren për të ekzekutuar të njëjtin bllok kodi disa herë, përderisa të plotësohet një kusht i caktuar. Në PHP ekzistojnë llojet e mëposhtme të cikleve:

**while** – përsërit një bllok të caktuar të kodit për aq kohë sa kushti i specifikuar të jetë i saktë  
**do...while** – përsërit një bllok të caktuar të kodit njëherë, dhe pastaj përsërit ciklin për aq kohë sa kushti i specifikuar të jetë i saktë  
**for** – përsërit një bllok të caktuar për një numër të specifikuar të herëve  
**foreach** – përsërit një bllok të caktuar të kodit për çdo element në një varg

### PHP – while Loop

Sintaksa

```
while (condition is true) {  
    code to be executed;  
}
```

1. Shkruani një program në PHP që printon numrat nga 1 deri në 5.

Shpjegim: `$x = 1;` - inicializon numëruesin e ciklit (`$x`), dhe vendos vlerën fillestare në 1  
`$x <= 5;` - vazhdon ciklin për aq kohë sa `$x` është më e vogël ose e barabartë me 5  
`$x++;` - rrit vlerën e numëruesit të ciklit për 1 për secilin iterim

2. Shkruani një program në PHP që numëron deri në 100 nga dhjetë numra (vlera).

### PHP – do while Loop

Sintaksa

```
do {  
    code to be executed;  
} while (condition is true);
```

3. Rishkruani shembullin 1, tani duke përdorur ciklin `do...while`.

Ky shembull së pari vendos një variabël `$x` në vlerën 1 (`$x = 1`). Pastaj, cikli `do while` do të printojë një rezultat (dalje) dhe pastaj inkrementon variablën `$x` për 1. Pastaj kontrollohet nëse plotësohet kushti i specifikuar (nëse `$x` është më i vogël ose i barabartë me 5?) dhe cikli do të vazhdojë të ekzekutohet për aq kohë sa variabla `$x` është më e vogël ose e barabartë me 5.

Mbani në mend: Në një cikël `do...while`, kushti testohet pasi të ekzekutohen deklaratimet brenda ciklit. Kjo nënkupton se cikli `do...while` do të ekzekutojë deklaratimet e tij të paktën një herë, edhe nëse kushti nuk është i saktë.

4. Përsërit detyrën 3, ashtu që vlera e variablës `$x` të vendoset në 6, ekzekutohet cikli dhe pastaj kontrollohet kushti.

## PHP – for Loop

### Sintaksa

```
for (init counter; test counter; increment counter) {  
    code to be executed for each iteration;  
}
```

### Parametrat:

*init counter*: inicializon vlerën e numëruesit të ciklit

*test counter*: vlerësohet për çdo iterim të ciklit. Nëse evaluohet në TRUE, cikli vazhdon. Nëse evaluohet në FALSE, cikli përfundon.

*increment counter*: rrit vlerën e numëruesit të ciklit

5. Shkruani një program në PHP që printon numrat nga 0 deri në 10.

Shpjegim: `$x = 0;` - inicializon numëruesin e ciklit (`$x`), dhe vendos vlerën startuese në 0

`$x <= 10;` - vazhdon ciklin derisa `$x` është më i vogël ose i barabartë se 10

`$x++` - rrit vlerën e numëruesit të ciklit për 1 për çdo iterim.

6. Shkruani një program në PHP që numëron në 100 nga dhjetë numra.

`$x = 0;` - inicializon numëruesin e ciklit (`$x`), dhe vendos vlerën fillestare në 0

`$x <= 100;` - vazhdon ciklin për aq sa `$x` është më i vogël ose barabartë me 100

`$x+=10` – rrit vlerën e numëruesit të ciklit për 10 për çdo iterim

## PHP – foreach Loop

Cikli `foreach` punon vetëm në vargjet, dhe përdoret për të iteruar nëpër secilin çift çelës/vlerë në një varg.

### Sintaksa

```
foreach ($array as $value) {  
    code to be executed;  
}
```

Për çdo iterim të ciklit, vlera e elementit aktual të vargut i caktohet variablës `$value` dhe pointeri i vargut zhvendoset për një, derisa të arrijë në elementin e fundit të vargut.

7. Shkruani një program në PHP që inicializon një varg me vlera të ndryshme (`$colors`) dhe printon vlerat e vargut të dhënë.
8. Shkruani një program në PHP që inicializon një varg me vlera të ndryshme (`$age`), dhe i ruan në çift çelës / vlerë. Printoni edhe çelësin dhe vlerën e vargut të dhënë.