

Dr. Holló Csaba, SZTE TTIK, Szoftverfejlesztés Tanszék

Programozási alapismeretek kurzus

Olvasási idő:  
5 perc



---

## Alapvető programozásnyelvi elemek C-ben és PHP-ban

### IV. Vezérlések

**A lecke célja** A vezérlések működésének jobb megértése.

**Tudás** A C-ben levő ciklusokon kívül ismeri a más nyelvekben gyakran használt foreach ciklust is.  
Összehasonlítja a ciklusokat.

**Képesség** A ciklusokat kölcsönösen átírja egymásra és megfelelően használja.

---

## While és Do-while

- Mindkét nyelvben azonosan működnek.
- A jobb megértés érdekében gondolkozzunk el azon, hogy ha az egyik nyelvben csak **while**, a másikban csak **do-while** lenne, akkor hogyan lehetne egymásra átírni?

**while**: mindaddig amíg teljesül egy feltétel, végezzük a cselekvést  $\leftrightarrow$  először a feltételt nézzük meg

**do-while**: végezzük a cselekvést a feltétel hamissá válásáig  $\leftrightarrow$  először cselekszünk, majd megnézzük, hogy csinálhatjuk-e tovább

# While és Do-while

```
while (not edge) {  
    run();  
}
```

```
do {  
    run();  
} while (not edge);
```



## While → Do-while

Technikailag a különbség az, hogy ha már először hamis a feltétel, akkor a ciklusmagot:

- a **while** egyszer sem végzi el
- a **do-while** egyszer elvégzi.

Hogyan tudjuk a **while**-t **do-while**-al helyettesíteni?

- ❖ Csak akkor végezzük el a **do-while**-t, ha kezdetben teljesül a feltétel.
- ❖ Utána mindkét ciklus csak akkor folytatja, ha továbbra is teljesül a feltétel.

## Do-while ↔ While

```
while(F){  
    A;  
}
```



```
if(F){  
    do{  
        A;  
    }while(F);  
}
```

Hogyan tudjuk a **do-while**-t **while**-al helyettesíteni?

Egyszer mindenképpen elvégezzük a ciklusmagot,  
utána a két ciklus ugyanúgy működik.

```
do{  
    A;  
}while(F)
```



```
A;  
while(F){  
    A;  
}
```

## Foreach

Tömbben vagy objektumban tárolt értékhalmoz minden elemére végrehajtja a ciklusmagot.

A ciklus ismétléseinek száma alapvetően nem a ciklusmag tartalmától, hanem a halmoz elemeinek számától függ.

Természetesen **break**-el itt is ki lehet ugrani a ciklusból, de ebben az esetben nem a **foreach** normál működését kapjuk, hiszen akkor már nem hajtódik végre minden elemre.

❖ [További leírások a foreach ciklusról PHP-ben.](#)

## Kérdések, feladatok

1. Tekintsük át a for ciklus működését az előadás diák alapján, illetve a [megabyte.hu for ciklusról szóló](http://megabyte.hu/for_ciklusról_szóló) oldalán, majd írjuk le, hogy hogyan tudjuk a **for**-t átírni **while**-ra és **do-while**-ra?
2. Írjuk át a **while** ciklust **for** ciklusra!
3. Írjuk át a **do-while** ciklust **for** ciklusra!
4. Hogyha tudjuk, hogy a **while** feltétele először mindig igaz lesz, akkor **do-while**-ra történő átíráskor a **do-while**-t bele kell-e tenni **if**-be?

## Kérdések, feladatok megoldásai

1. Legyenek ut1, ut2 utasítások.

```
for(ut1;F;ut2){  
    A;  
}
```

```
ut1;  
while(F){  
    A;  
    ut2;  
}
```

```
ut1;  
if(F){  
    do{  
        A;  
        ut2;  
    }while(F);  
}
```

2. `while(F){ A; }` → `for(;F;){ A; }`

3. `do{ A; } while(F);` → `A; for(;F;){ A; }`

4. NEM, mert tudjuk az if eredményét.



## Kérdések, feladatok

5. Milyen ciklusnak felel meg az „addig jár a korszó a kútra, míg el nem török” közmondás?

**Megoldás.** A megfogalmazás azt sugallja, hogy kezdetben ép a korszó, ezért legalább egyszer elmegy a kútra, ezért do-while.

6. Milyen ciklusnak felel meg a „Ha a medve megszagolta a mézet, abban nem hagyja.” közmondás?

**Megoldás.** while(szagolja){szagolja;} → végtelen

7. Tegyük egy kitekintést a ciklusok különböző programozási nyelvekben való megvalósítására a [Berzsenyi Dániel Gimnázium Programozás jegyzet](#) oldalán.

# EFOP-3.4.3-16-2016-00014

## AP1 HALLGATÓI DIPLOMA-SZERZÉST SEGÍTŐ SZOLGÁLTATÁSOK

Jelen tananyag a  
Szegedi Tudományegyetemen  
készült az  
Európai Unió támogatásával.  
Projekt azonosító:  
EFOP-3.4.3-16-2016-00014



**SZÉCHENYI** 2020

**Európai Unió**  
Európai Szociális  
Alap



**BEFEKTETÉS A JÖVŐBE**