# Vezérlési szerkezetek

## Github link

<https://github.com/R1chard2001/os1_vezerlesi_szerkezetek>

## Elágazások

### if

**#!/bin/bash**  
**…**  
**if [** *<kifejezés>* **]  
then**  
**…**

**elseif [** *<kifejezés>* **]**

**…  
fi**

**#!/bin/bash**  
**…**  
**if [** *<kifejezés>* **]  
then**  
**…**

**else**

**…  
fi**

**#!/bin/bash**  
**…**  
**if [** *<kifejezés>* **]  
then**  
**…  
fi**

**#!/bin/bash**  
**…**  
**if [** *<kifejezés>* **]; then**  
**…  
fi**

### case

**#!/bin/bash**  
**…**  
**case** *<változó>* **in**

*<eset>***) … ;;**

*<másik eset>***)**

**…**

**;;**

**\*) … ;;** *#ez az alapértelmezés*  
**esac**

**…**

## Ciklusok

### for

Olyan mint a C#-ban lévő foreach.

**#!/bin/bash**  
**…**  
**for** *<változónév>* **in** *<változó>*; **do**  
**…  
done**

**…**

**#!/bin/bash**  
**…**  
**for** *<változónév>* **in** *<változó>*

**do**  
**…  
done**

**…**

### while

**#!/bin/bash**  
**…**  
**while read** *<változónév>*

**do**  
**…  
done**

**…**

**#!/bin/bash**  
**…**  
**while [** *<kifejezés>*  **]**

**do**  
**…  
done**

**…**

Ha a while-t read segítségével használjuk, akár saját magunk is készíthetünk szűrőket.

## egyéb használt parancsok

### seq *<mettől> <milyen lépésben> <meddig>*

Számokat kiírja a megadott paramétereket használva.

### [

Ez a program segíti kiértékelni a logikai kifejezést. Sok hasznos kapcsolója van. (**man [** segítségével megtekinthetjük melyek ezek.)

### break

Segítségével kiléphetünk a ciklusból.

### exit [szám]

Segítségével kiléptetjük a programot.

Ha adunk neki számot paraméterül, akkor a program kilépésének eredményét is megadhatjuk. (Ezt program lefutása utána a **$?** változó segítségével ellenőrizhetjük.) Alapértelmezetten 0 a szám értéke.