

## Vezérlési szerkezetek

A vezérlési szerkezetek a programozás alapvető elemei, amelyek meghatározzák a program végrehajtási sorrendjét. Három fő típusa van: **szekvencia**, **szelekció**, és **iteráció**.

### 1. Szekvencia

A szekvencia egy sorozatban végrehajtott utasításokat jelent. Ez a program legegyszerűbb vezérlési szerkezete, amelyben az utasítások egymás után hajtódnak végre, az első utasítástól az utolsóig.

### 2. Szelekció - elágazás

A szelekció (vagy elágazás) segítségével feltételek alapján dönthetünk, hogy melyik utasítást vagy utasítássorozatot hajtsuk végre. Ez lehetőséget ad a programnak, hogy különböző műveleteket végezzen különböző feltételek alapján.

### 3. Iteráció - ciklus

Az iteráció (vagy ciklus) segítségével egy utasítássorozatot többször is végrehajthatunk. Ez hasznos, ha ismétlődő feladatokat kell elvégezni, például egy lista elemein végigmenni vagy egy számítást többször végrehajtani.

## Elágazások

### 1. Egyirányú elágazás

Az egyirányú elágazásban a `if` kulcsszót használjuk a feltétel megadására. A feltétel után kettőspont (`:`) következik, majd a blokkja egy szinttel beljebb kezdődik.

#### Szintaxis:

```
if feltétel:  
    # utasítás(ok), ha a feltétel igaz
```

#### Példa:

```
x = 10  
if x > 5:  
    print("x nagyobb mint 5")
```

#### Magyarázat:

- `if x > 5::` Ez a feltétel, amelyet a program ellenőriz.
- `print("x nagyobb mint 5"):` Ez az utasítás csak akkor hajtódik végre, ha a feltétel igaz.

## 2. Kétirányú elágazás

A kétirányú elágazásban a `if` és `else` kulcsszavakat használjuk. Ha a `if` feltétel igaz, az első blokk hajtódik végre. Ha a feltétel hamis, az `else` blokk hajtódik végre.

### Szintaxis:

```
if feltétel:  
    # utasítás(ok), ha a feltétel igaz  
else:  
    # utasítás(ok), ha a feltétel hamis
```

### Példa:

```
x = 10  
if x > 5:  
    print("x nagyobb mint 5")  
else:  
    print("x kisebb vagy egyenlő mint 5")
```

### Magyarázat:

- `if x > 5::` Ez a feltétel, amelyet a program ellenőriz.
- `print("x nagyobb mint 5"):` Ez az utasítás hajtódik végre, ha a feltétel igaz.
- `else::` Ez a blokk hajtódik végre, ha a feltétel hamis.
- `print("x kisebb vagy egyenlő mint 5"):` Ez az utasítás hajtódik végre, ha a feltétel hamis.

## 3. Többirányú Elágazás

A többirányú elágazásban a `if`, `elif` (ami az "else if" rövidítése) és `else` kulcsszavakat használjuk. A program sorban ellenőrzi a feltételeket, és az első igaz feltételhez tartozó blokkot hajtja végre. Ha egyik feltétel sem igaz, az `else` blokk hajtódik végre (ha van).

### Szintaxis:

```
if feltétel1:  
    # utasítás(ok), ha a feltétel1 igaz  
elif feltétel2:  
    # utasítás(ok), ha a feltétel1 hamis és feltétel2 igaz  
else:  
    # utasítás(ok), ha mindkét feltétel hamis
```

### Példa:

```
x = 10  
if x > 10:  
    print("x nagyobb mint 10")  
elif x == 10:  
    print("x egyenlő 10-zel")  
else:  
    print("x kisebb mint 10")
```

**Magyarázat:**

- `if x > 10::` Ez az első feltétel, amelyet a program ellenőriz.
- `print("x nagyobb mint 10"):` Ez az utasítás hajtódik végre, ha az első feltétel igaz.
- `elif x == 10::` Ha az első feltétel hamis, a program ellenőrzi ezt a második feltételt.
- `print("x egyenlő 10-zel"):` Ez az utasítás hajtódik végre, ha a második feltétel igaz.
- `else::` Ha egyik feltétel sem igaz, ez a blokk hajtódik végre.
- `print("x kisebb mint 10"):` Ez az utasítás hajtódik végre, ha egyik feltétel sem igaz.