

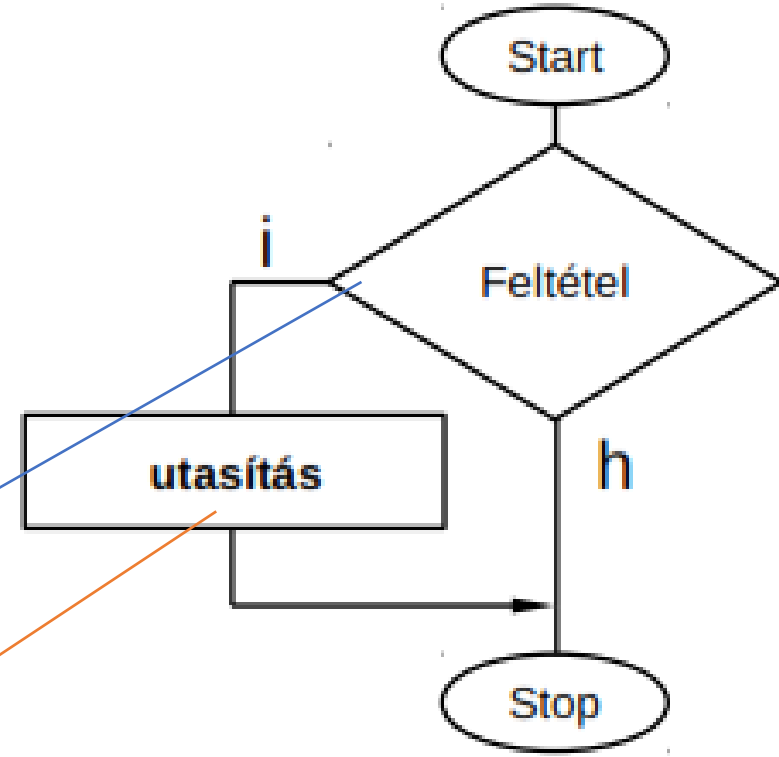
ELÁGAZÁSOK

if – elif – else

Készítette: Vastag Attila

2016

Gyakran előfordul, hogy meg kell vizsgálnunk egy állítást, és attól függően, hogy igaz vagy hamis, a programnak más-más utasítást kell végrehajtania. Ilyen esetekben elágazást használunk az **if** (ha) kulcsszó segítségével:

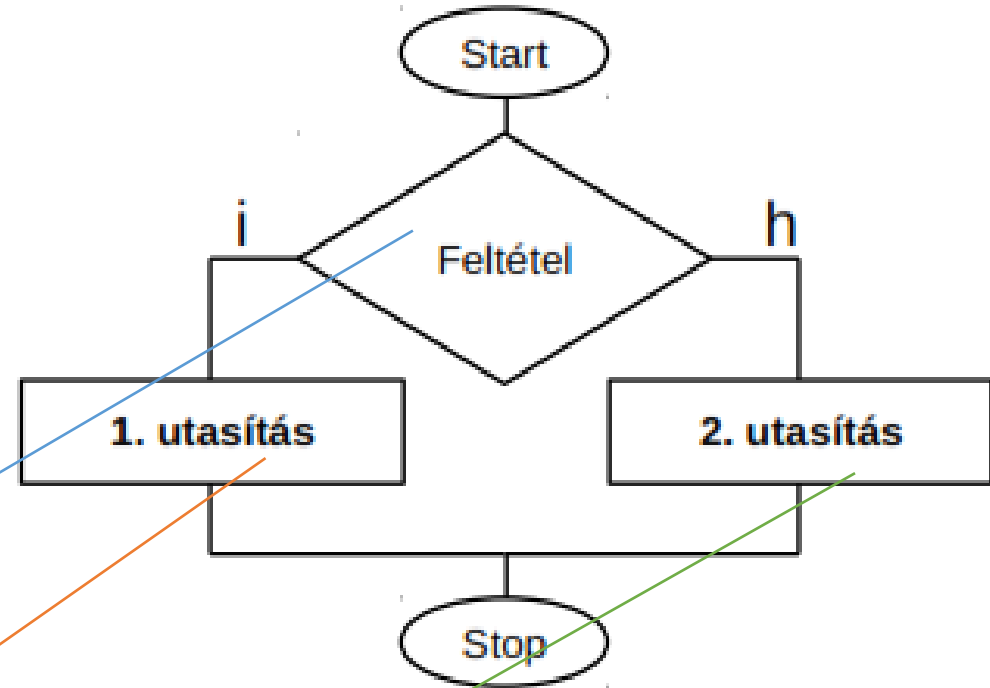


x: int = 10

if (x == 10):

print("x értéke 10")

Természetes az igény arra is, hogy azt a helyzetet is kezelni tudjuk, amikor x értéke nem tíz. Ilyenkor használjuk az *else* ágat:



```
x: int = 11
```

```
if (x == 10):  
    print("x értéke 10")  
else:  
    print("x értéke nem 10")
```

Az *else* szerkezet akkor lép életbe, ha a hozzá kapcsolódó feltétel(ek) nem igaz(ak).
Önmagában *else* ág nem állhat (nem is lenne sok értelme).

A fenti helyzetben írhattuk volna ezt is:

```
x: int = 11
```

```
if (x == 10):  
    print("x értéke 10")
```

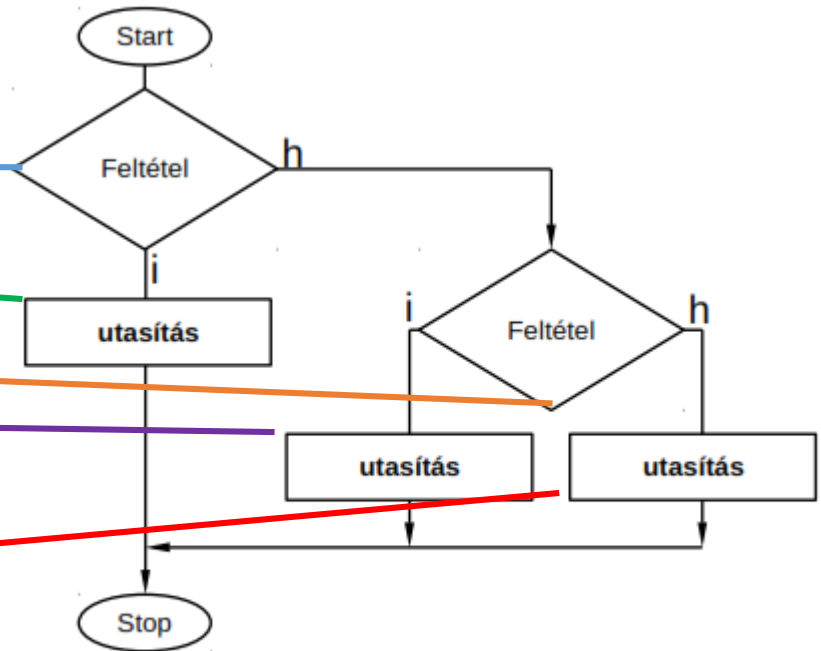
```
if (x != 10):  
    print("x értéke nem 10")
```

Ez a program pontosan ugyanazt csinálja, mint az előző, de van egy nagy különbség a kettő között: mindkét feltételt ki kell értékelnie a programnak, hiszen két különböző szerkezetről beszélünk (ez egyúttal azzal is jár, hogy a feltételtől függően mindkét állítás lehet igaz, és nem valószínű, hogy a várt eredményt kapjuk logikailag).

Arra is van lehetőségünk, hogy több feltételt is megvizsgáljunk, ekkor **elif** –et használunk:

```
x: int = 13
```

```
if (x == 10):  
    print("x értéke 10")  
elif (x == 12):  
    print("x értéke 12")  
else:  
    print("x értéke nem 10 vagy 12")
```



A program az első olyan ágot fogja végrehajtani, amelynek a feltétele teljesül (vagy ha egyik feltétel sem bizonyult igaznak, akkor az **else** ágot – ha adtunk meg ilyet, de nem feltétlenül szükséges).

Egy elágazásban pontosan egy darab **if**, bármennyi **elif** és pontosan egy **else** ág lehet. Egy elágazáson belül is írhatunk elágazást.

A valóságban legtöbbször olyan esetekkel találkozunk, ahol az igaz ágon, vagy a hamis ágon nem csak egy utasítást kell végrehajtanunk.

Ezeket az utasításokat blokkokba írjuk ***tabulátoros*** behúzással.

if (*feltétel*):

utasítás 1

utasítás 2

utasítás n

utasítás blokk

elif (*feltétel*):

utasítás 1

utasítás 2

utasítás n

utasítás blokk

else:

utasítás 1

utasítás 2

utasítás n

utasítás blokk

Feladatok

1. Egy beolvasott számról döntse el a program hogy nagyobb e 0-nál, ha igen írja is ki!
2. A képernyőre írjuk ki hogy „pozitiv,, ha a beolvasott szám nagyobb vagy egyenlő nulával, ha viszont kisebbbb akkor hogy „negativ,,.
3. Egy beolvasott számról döntse el a program hogy -30 és 40 között van-e!
4. Két beolvasott szám közül írassuk ki a nagyobbikat!
5. Rendezünk sorba 2 egész számot! (írja ki a kisebbet, majd a nagyobbbat)

Feladatok

6. Rendezzünk sorba 3 egész számot!

7. Olvassunk be egy számot konzolról és állapítsuk meg, hogy osztható e 5-el.

8. Olvassunk be egy számot konzolról és állapítsuk meg, hogy osztható e 4-el és 6-al is.

9. Olvassunk be két egész számot, x és y , konzolról és állapítsuk meg, hogy az y osztója e az x -nek.

10. Olvassunk be egy egész számot. Ha a szám osztható 2-vel, írjuk ki hogy „BIZ”, ha osztható 3-al, írjuk ki hogy „BAZ” és ha osztható 2-vel és 3-al is akkor írjuk ki, hogy „ZIZI”.

11. Olvassunk be egy számot, majd e számról állapítsuk meg, hogy páros/páratlan, pozitív/negatív és osztható-e öttel.

Feladatok

12. Olvassunk be konzolról egy számot, majd állapítsuk meg, hogy ez a szám 10 és 20 közt van-e, vagy -10 és -20 közt.

13. Olvassunk be konzolról egy számot, majd állapítsuk meg, hogy ez a szám 0 és 9 közt van-e, vagy 10 és 99 , vagy 100 és 999 közt.

Ha 0 és 9 közt van írjuk ki, hogy egyjegyű szám.

Ha 10 és 99 közt van írjuk ki, hogy kétegyű szám.

Ha 100 és 999 közt van írjuk ki, hogy háromegyű szám.

14. Olvassunk be konzolról három számot : x, y és z. Állapítsuk meg, hogy az X osztható e Y-nal, osztható e Z-vel és osztható e mindkettővel!