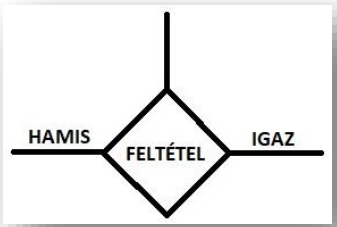
08. FELTÉTELES ELÁGAZÁS -IF

Programozásban fontos az utasítások sorrendje. Ha ugyanazon utasítások sorrendjét felcseréljük, akkor más lesz a végkifejlet. A **vezérlési szerkezet**ek alkalmazásával dönthetjük el, hogy melyik utasítás következzen.
Az eddigi programjainkban **sorrendi végrehajtást (szekvenciát)** alkalmaztunk.
Mostantól pedig **elágazásokat (szelekciókat)** fogunk használni.
Ha egy feltételtől függően a program többféleképpen folytatódhat, akkor **elágazást alkalmazunk**.
A feltétel egy logikai érték, tehát lehet igaz (True), vagy hamis (False). Attól függően, hogy egy vizsgálatra adott válasz igaz, vagy hamis, különböző irányban folytatódik a program.



Öt fajta elágazást fogunk megnézni:

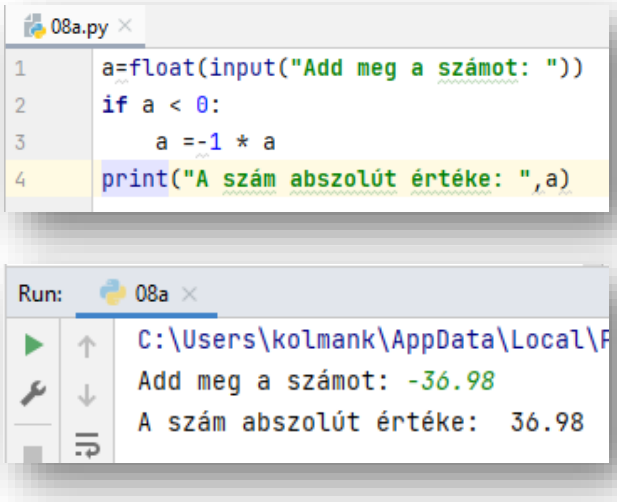
- **Elágazás egy irányban (if)**
- **Elágazás két irányban (if-else)**
- **Többszörös elágazás (if-elif-else)**
- **Egymásba ágyazott elágazás (if-else-if-else)**
- **Elágazás összetett feltétellel (not, and, or)**

Elágazás egy irányban (if)

Egyirányú elágazásról akkor beszélünk, ha csak az igaz ágat hozzuk létre!
Mondatszerű leírás: **Ha <feltétel> akkor <utasítások>**
Elágazás vége
Az előző mondat értelmezése a következő: a feltétel teljesülése esetén az „akkor mögötti utasítások végrehajtódnak. Ha a feltétel nem teljesül, akkor ezt a feladatrész átugorva folytatódik a program végrehajtása.

(08a.py)
Hozzunk létre egy új python fájlt, a neve legyen 08a.py!

A feladatban számítsuk ki egy bekért szám abszolút értékét. Írjuk meg az algoritmust úgy, hogy ha a szám nem negatív, akkor semmit nem kell tenni a számmal, ha viszont negatív, akkor a számnak kell venni a mínusz egyszeresét. (Most nem az abs() függvényt használjuk.)



A nagyon rövid program **mondatszerű leírás**a a következő:

(1) Be: a
(2) Ha a<0 akkor
(3) a = -1*a
(4) Ki: a

- A feladatot úgy oldjuk meg, hogy először bekérünk egy valós számot, ami lehet pozitív, vagy negatív.
- Majd indítunk egy **„if" feltételt**, ahol megvizsgáljuk, hogy az „a" változó kisebb-e mint nulla. Mert, ha igen a válasz a feltételre, akkor végrehajtóik a következő (automatikusan beljebb kezdődő) következő sor.
- A feltétel után a Pythonban : (kettőspon) karaktert kell elhelyezni. **Minden vezérlési szerkezetnél meg kell adni a :-ot.** Tehát az if elágazás miatt kell a „:"-ot beírni!
- Ha a válasz nem, akkor átugorja a beljebb kezdődő utasításoka, és csak egyszerűen kiírja az eredeti számot!

Elágazás két irányban (if-else)

A kétirányú elágazásnál, a feltételtől függően mind a két irányba különböző utasítások hajtódnak végre! Tehát a hamis ágban is lesznek utasítások.
Mondatszerű leírása a következő:

Ha <feltétel> akkor <utasítások1>
különben <utasítások2>
Elágazás vége

Az előzőekben leírtak értelmezése szerint , ha a megadott feltételre igen a válasz, akkor az „utasítások1” hajtódik végre, ha pedig nem a válasz a feltételre, akkor az <utasítások2> hajtódik végre! Ezután az „Elágazás vége” után folyatódik a program.

(08b.py)
Hozzunk létre egy új python fájlt, a neve legyen 08b.py!

Számítsuk ki egy bekért szám négyzetgyökét. Ha a szám negatív, akkor írassuk ki, hogy a feladat nem végrehajtható!

- A programban az első sorban elérhetővé tesszük az sqrt() függvényt a „from math import *” utasítással!
- Bekérjük a valós számot!
- Létrehozunk egy elágazást if utasítással, az utána következő feltétellel és a kettősponttal!
- Az „igaz” ágban kiíratjuk a szöveget! Fontos, hogy a program automatikusan beljebb kezdi az ág utasításait. Ezen nem szabad változtatni!
- Ha véletlenül nem kezdődik beljebb, akkor nekünk kell Tabot nyomi, vagy négy szóközt!
- Aztán jön egy „else” (különben) ág, melyben a hamis válasz után történő utasításokat helyezzük. Jelen esetben a gyökvonást!
- Futtassuk a programot, mind a két eshetőséget teszteljük. Azesetleges hibákat javítsuk!

```
from math import *
a=float(input("Add meg a számot: "))
if a < 0:
    print("Nem végezhető el a művelet!")
else:
    print("A szám gyöke: %.2f" % (sqrt(a)))
```

```
Run: 08b
C:\Users\kolmank\AppData\Loca
Add meg a számot: -35.14
Nem végezhető el a művelet!
```

```
Run: 08b
C:\Users\kolmank\AppData\Lo
Add meg a számot: 36.47
A szám gyöke: 6.04
```

A Pythoon programozási nyelvben a következő módon használjuk az összehasonlító relációs jeleket!

Összehasonlítás	Mondatszerű leírás Algoritmusok	Python
nagyobb	a>b	**a>b**
kisebb	a<b	**a<b**
nagyobb egyenlő	a>=b	**a>=b**
kisebb egyenlő	a<=b	**a<=b**
egyenlő	a=b	**a==b**
nem egyenlő	a<>b	**a!=b**

Ezeket nagyon meg kell jegyezni, mert sokszor fogjuk használni!

Elágazás több irányban (if-elif-else)

Ha kettőnél több kimenet van, akkor alkalmazhatunk „elif” ágakat is. Tehát több feltételt vizsgálhatunk!
Nagyon fontos viszont a feltételek sorrendje, mert **fentről lefelé indulva vizsgálja a feltételeket**, és **az első igaz állításra kilép**, és az **elágazás után folytatja**! Az első if feltétel után elif feltételekkel adhatjuk meg az összehasonlítás lehetőségeit, majd az utolsó ág itt is a „különben” else ág!

(08c.py)
Hozzunk létre egy új python fájlt, a neve legyen 08c.py!

Ebben a feladatban kérjünk be egy számot, melyről állapítsuk meg, hogy pozitív, nulla, vagy negatív!

- Bekérünk egy valós számot!
- Az első (if) vizsgálatnál megnézzük, hogy a szám nagyobb-e mint 0! Ha igen, akkor végrehajtódik az első ptint() utasítás.
- A második vizsgálat már az „elif” utasítás mögött van, ahol azt vizsgáljuk, hogy egyenlő-e nullával! Mert ha igen, akkor végrehajtódik a második print() utasítás.
- A különben ágban pedig már csak a negatív számok maradtak, nem kell vizsgálni semmit, csak a szöveget kiíratni!
- Teszteljük mind a három lehetőségre!

```
08c.py
1  a=float(input("Addj meg egy számot: "))
2  if a>0:
3      print("A szám pozitív.")
4  elif a==0:
5      print("Nulla.")
6  else:
7      print("A szám negatív")

Run:  08c
C:\Users\kolmank\AppData\Loca
Addj meg egy számot: -26.13
A szám negatív
```

Egymásba ágyazott feltételvizsgálat (if-else-if-else)

Az előző feladatot megadhatjuk másféleképpen is. Nem „elif” utasítás használatával, hanem az if-else egymásba ágyazásával.

(08d.py)
Hozzunk létre egy új python fájlt, a neve legyen 08d.py!

A feladat ugyan az mint az előbb!

- Ebben az esetben is bekérünk egy valós számot!
- Csak if és else ágakat hozunk létre, a különböző lehetőségek végig vételével!
- Fontos, hogy amikor új fi ágat indítunk az utasításokat beljebb kell kezdeni!
- A minta programon látszik a tagolás. Ezt minden esetben be kell tartani!
- Teszteljük mind a három lehetőséget!

```
08d.py
1  a=float(input("Add meg a számot: "))
2  if a>0:
3      print("Pozitív.")
4  else:
5      if a==0:
6          print("Nulla.")
7      else:
8          print("Negatív")

Run:  08d
C:\Users\kolmank\AppD
Add meg a számot: 0
Nulla.
```

Elágazás összetett feltétellel (not, and, or)

Az összetett feltételek részfeltételekből állnak, amelyeket a logikai műveletekkel tudunk egymás után helyezni. Az if feltétel után írjuk be a megfelelő képletet.
Ilyen feladat lehet például, hogy a változó értéke az 50 nél nagyobb, de 100-nál kisebb szám-e.
Logikai műveletek segítségével tudunk több logikai értékből egyet készíteni, összetett feltétellé összekapcsolni. Tehát az előző példát megnézve: (a>50 and (a<100)

A logikai műveleteket átismételve (True, False):

- **AND ➔ Akkor igaz, ha minden állítás igaz!**
- **OR ➔ Akkor igaz, ha legalább az az egyik állítás igaz!**
- **NOT ➔ Az ellenkezőjére változtatja az értéket!**

(08e.py)
Hozzunk létre egy új python fájlt, a neve legyen 08e.py!

Egy angol nyelvvizsgán az írásbelin 100, szóbelin 50 pontot lehet elérni. A vizsgán csak akkor mehetsz át, ha mind a két vizsgán elérted a minimum 60%-ot!
Készítsünk programot, amely bekér két pontszámot, majd eldönti, hogy átmentél e a vizsgán.

```
i=int(input("Add meg az írásbeli eredményét (0-100): "))
sz=int(input("Add meg a szóbeli eredményét (0-50): "))
iszaz=i/100
szszaz=sz/50
#print(iszaz,szszaz)
if (iszaz>0.6) and (szszaz>0.6):
    print("Sikeres a vizsga.")
else:
    print("Sikertelen a vizsga.")
```

```
Run: 08e
C:\Users\kolmank\AppData\Local\Programs\Pyt
Add meg az írásbeli eredményét (0-100): 81
Add meg a szóbeli eredményét (0-50): 44
Sikeres a vizsga.
```

- Először bekérjük az elért pontokat.
- Aztán kiszámoljuk a százalékokat, két változóba!
- Majd kezdjük a HA függvényt, megadjuk „és” feltétellel a vizsgálatokat, sikeres vizsga esetén.
- Az „else” ágban pedig a sikertelen vizsga esetén való kiíratást adjuk meg.

(08f.py)
Hozzunk létre egy új python fájlt, a neve legyen 08f.py!

Egy matematika dolgozat, négy nehéz feladatból áll. Ha bármelyik feladatot sikerült megoldani 75%-ra, akkor megkapja a legjobb jegyet! Minden feladat 20 pontos! Készíts programot a képernyőkép alapján! (A végén: „A dolgozat sikeres.” vagy a „A dolgozat sikertelen.” szöveg jelenjen meg!

```
Run: 08f
C:\Users\kolmank\AppData\Local\
Add meg az 1. pontszámát: 21
Add meg az 2. pontszámát: 35
Add meg az 3. pontszámát: 41
Add meg az 4. pontszámát: 49
A dolgozat sikeres.
```

Hasznos parancsok: **break, exit, quit**
Ha bármikor azt szeretnénk, hogy a program kilépjen az aktuális cikusból akkor, egyszerűen, minden paraméter nélkül használjuk a **break** parancsot!
Ha a programból is ki a karunk lépni, akkor használjuk az **exit()** vagy a **quit()** parancsokat.