

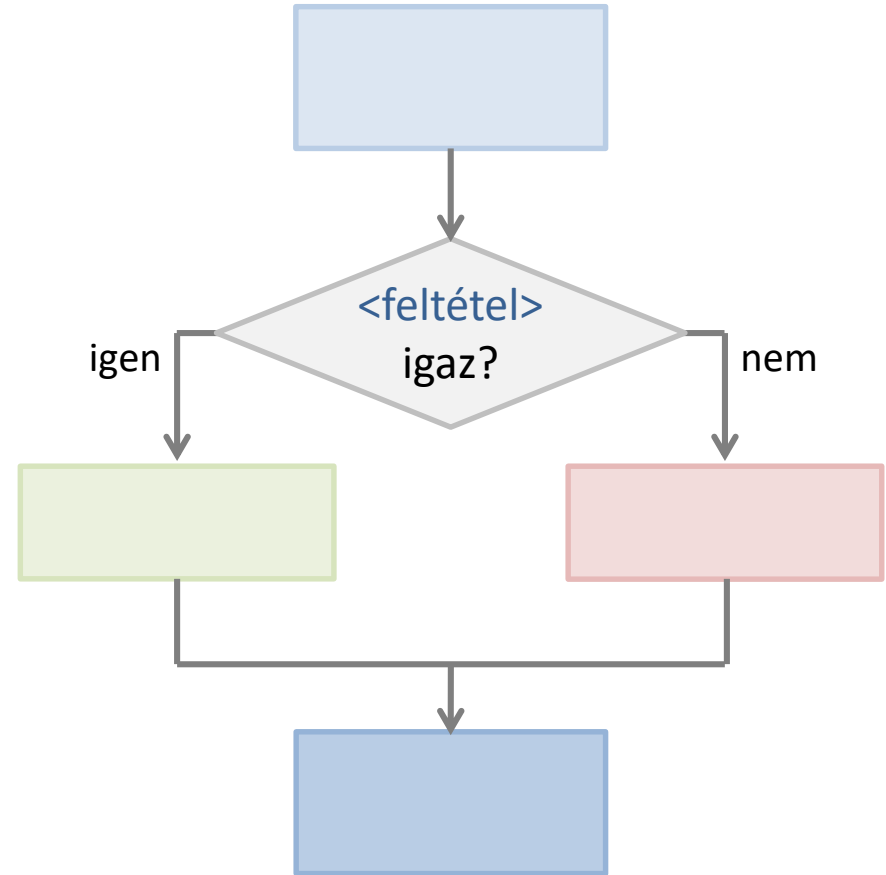


Vezérlési szerkezetek: feltételes elágazás és ciklusok

többszörös elágazás (if-elif-else)
kilépés while ciklusból (break), ciklus folytatása
(continue), és a while ciklus feltételéhez tartozó else ág
a for ciklus és a range() függvény
teknőc grafika (turtle modul)



Feltételes elágazás (már ismerjük!)





Többszörös elágazás

Módosítsd ezt a programot!

Módosítsd ezt a programot úgy, hogy három esetet különböztessen meg:

- ha a szám nagyobb, mint 100, akkor írja ki, hogy ez egy nagy szám,
- ha a szám kisebb, mint 10, akkor írja ki, hogy ez egy kicsi szám,
- egyébként írja ki, hogy a szám se nem kicsi, se nem nagy!

```
szam = int(input("Írj be egy számot! "))  
if (szam > 100):  
    print("Ez egy nagy szám.")  
else:  
    print("Ez egy kicsi szám.")
```





Többszörös elágazás

```
szam = int(input("Írj be egy számot! "))
if (szam > 100):
.....> print("Ez egy nagy szám.")
else:
    if (szam < 10):
.....> print("Ez egy kicsi szám.")
.....> else:
.....> print("Ez egy se nem kicsi, se nem nagy szám.")
```





Többszörös elágazás

- a többszörös elágazás kényelmesebben megoldható az `elif` (else-if) utasítással

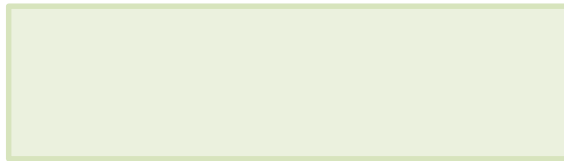
```
szam = int(input("Írj be egy számot! "))  
if (szam > 100):  
    .....> print("Ez egy nagy szám.")  
elif (szam < 10):  
    .....> print("Ez egy kicsi szám.")  
else:  
    .....> print("Ez egy se nem kicsi, se nem nagy szám.")
```





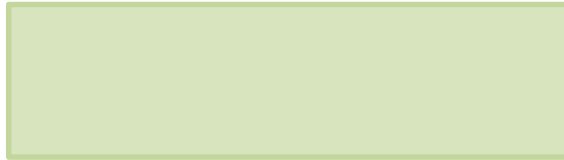
Többszörös elágazás

```
if <feltétel_1>:
```



Ez az utasításblokk akkor hajtódik végre ha **<feltétel_1> igaz**

```
elif <feltétel_2>:
```



Ez az utasításblokk akkor hajtódik végre ha **<feltétel_1> hamis, de <feltétel_2> igaz**

```
elif <feltétel_3>:
```



Ez az utasításblokk akkor hajtódik végre ha **<feltétel_1> és <feltétel_2> hamis, de <feltétel_3> igaz**

```
else:
```



Ez az utasításblokk akkor hajtódik végre ha **mindegyik <feltétel_> hamis**

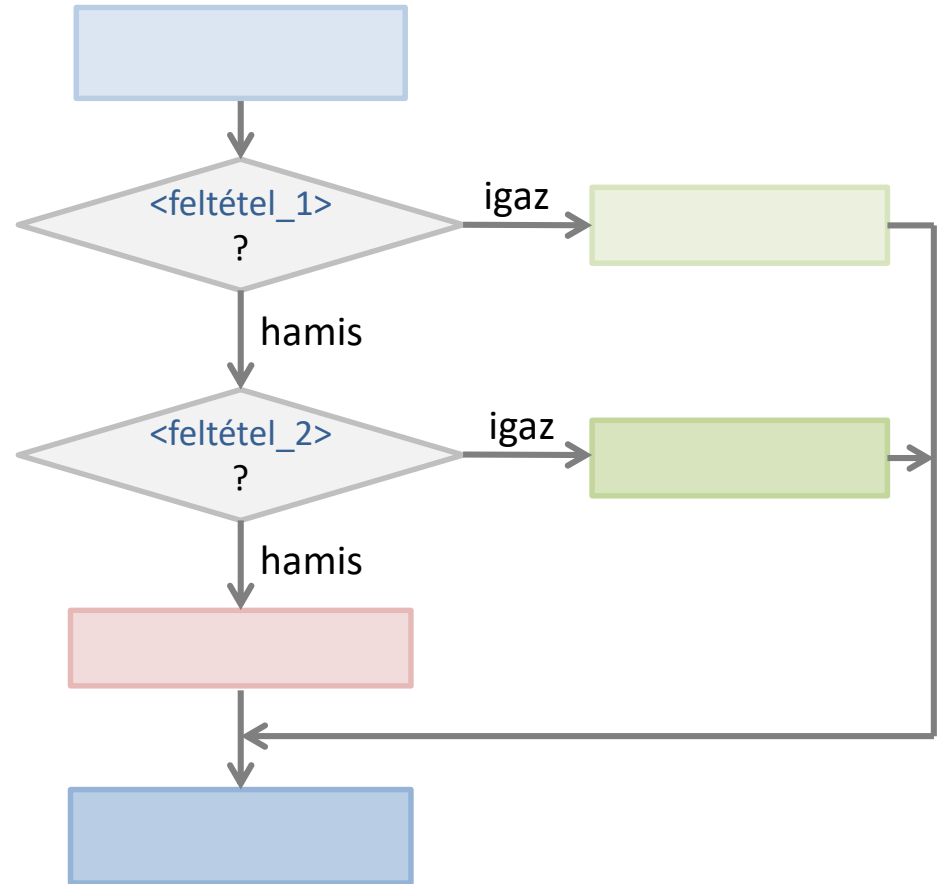




Többszörös elágazás

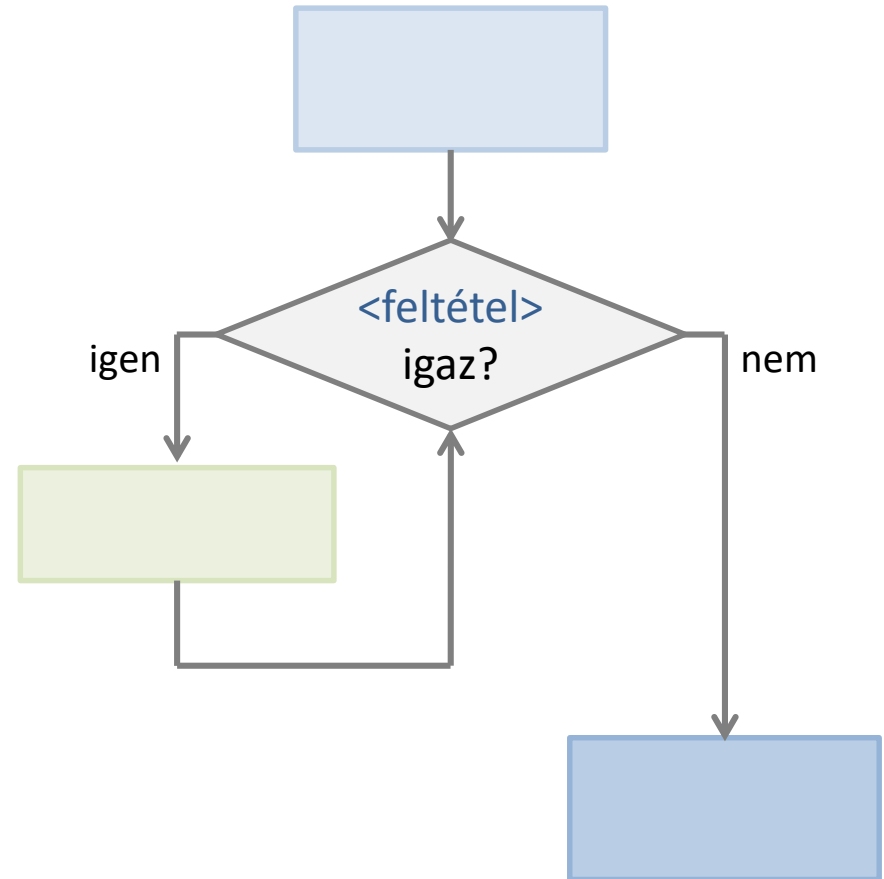
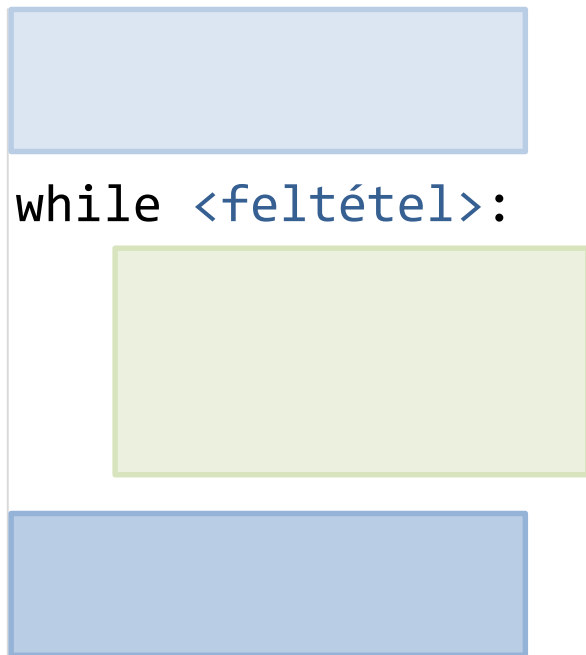
```

if <feltétel_1>:
    [green box]
elif <feltétel_2>:
    [green box]
else:
    [red box]
[blue box]
    
```





Ismétlés `while` ciklussal (már ismerjük!)





Emlékszel erre a programra?

```
from math import sqrt
szam = int(input("Írj be egy pozitív egész számot: "))
if (szam == 1):
    print("Definíció szerint nem prím.")
else:
    prim = True
    n = 2
    while (n <= sqrt(szam)):
        if (szam%n == 0):
            prim = False
            n = n + 1
    if (prim):
        print("Prím.")
    else:
        print("Nem prím.")
```

Ha találtunk egy osztót, akkor már tudjuk, hogy a szám nem lehet prím, és felesleges a további n-ek ellenőrzése. Jó lenne valahogy kilépni a while ciklusból...





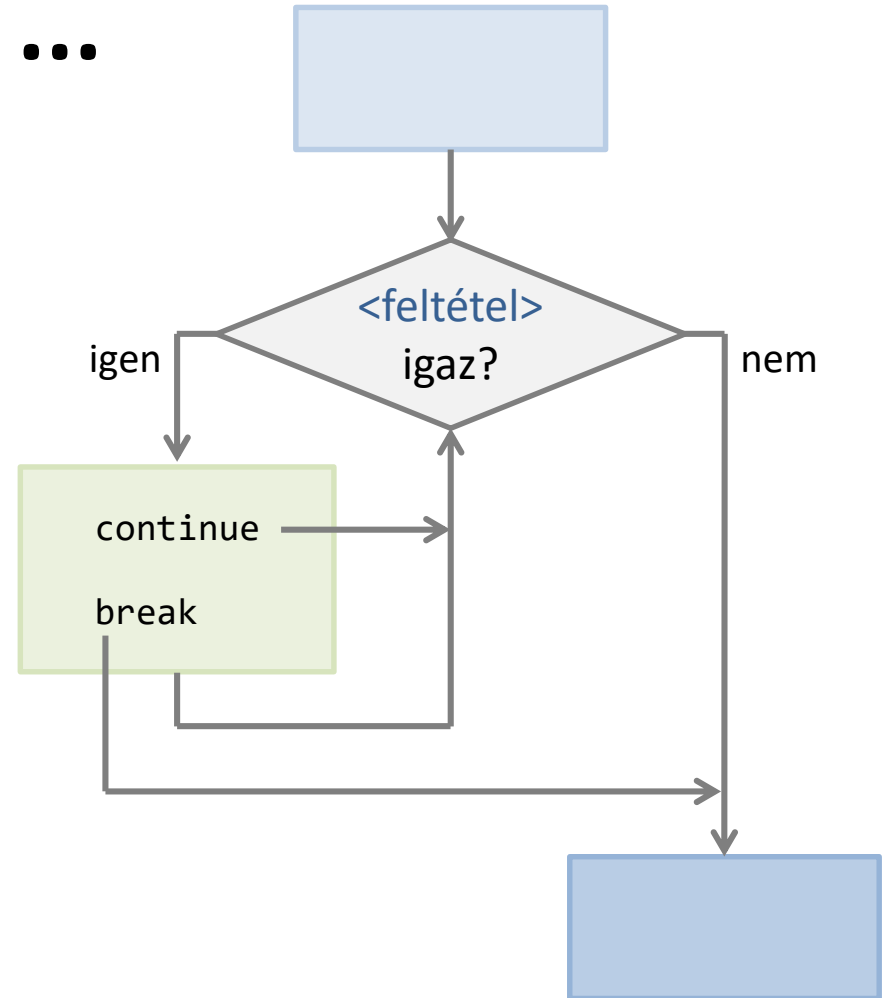
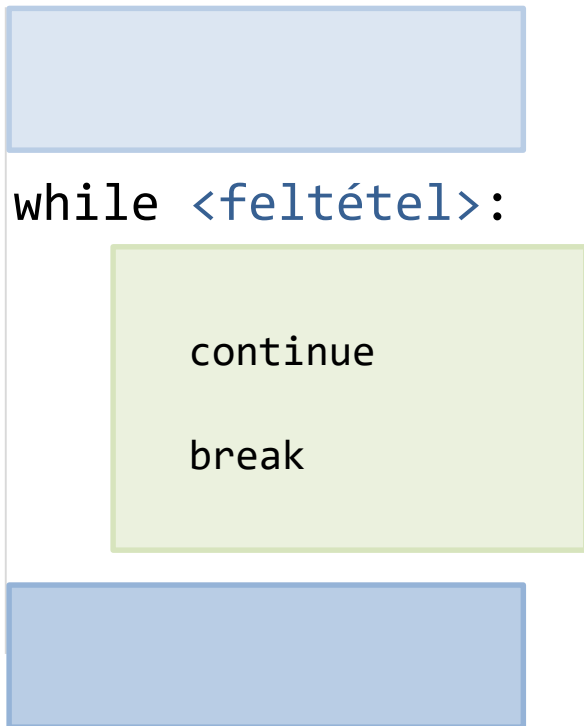
Kilépés a ciklusból a break utasítással

```
from math import sqrt
szam = int(input("Írj be egy pozitív egész számot: "))
if (szam == 1):
    print("Definíció szerint nem prím.")
else:
    prim = True
    n = 2
    while (n <= sqrt(szam)):
        if (szam%n == 0):
            prim = False
            break
        n = n + 1
    if (prim):
        print("Prím.")
    else:
        print("Nem prím.")
```



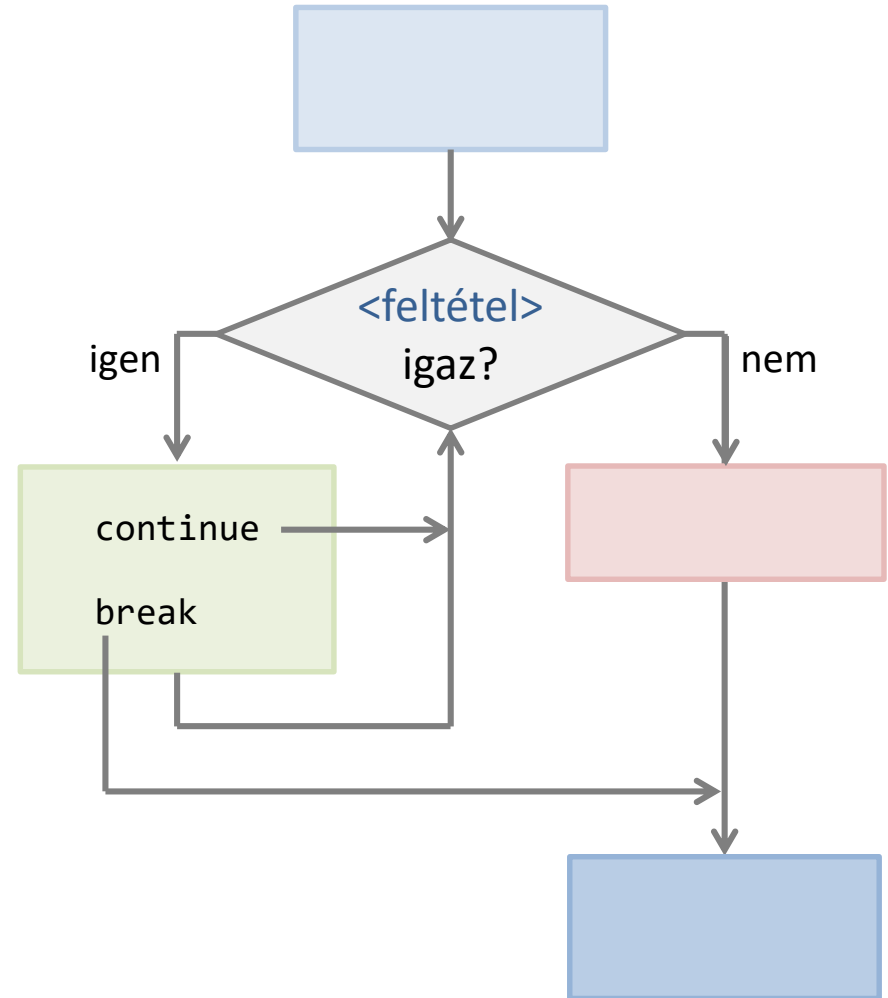
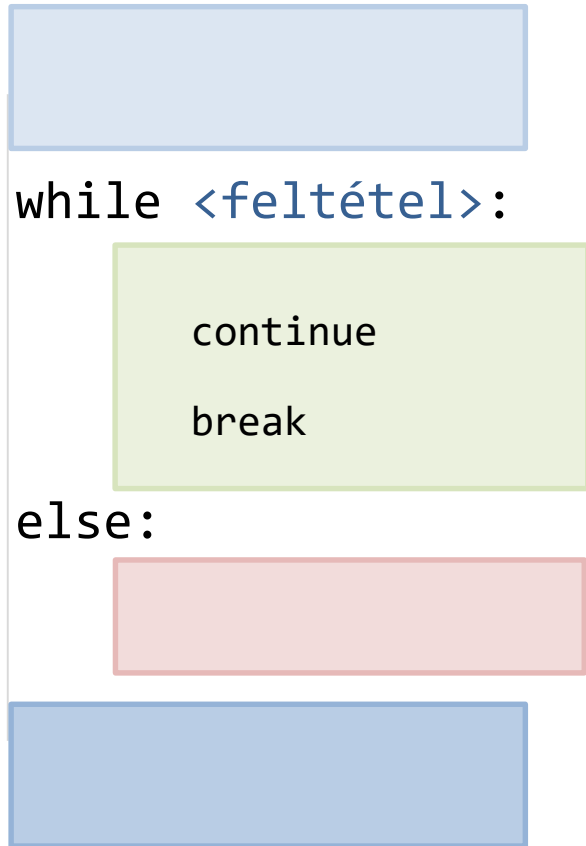


break, continue, ...





... és else





Gondoltam egy számra! version 2

Egészítsd ki az alábbi programot!

Emlékszel még a „Gondoltam egy számra!” programra? Most megfordítjuk a szerepeket: te gondolsz egy számra és a számítógép fogja kitalálni! Segítségül a program egy részét megadtuk, neked csak ki kell egészíteni azokkal az elemekkel, amiket most tanultunk:

- hármás elágazás (`if – elif – else`)
- `break` utasítás a `while` cikluson belül
- `while` ciklushoz tartozó `else` ág





```
print("Gondolj egy számra 1 és 100 között!")
input("Ha megvan, nyomd meg az ENTER-t!")

lepes = 0
h, H = 1, 101
while (h != H):
    tipp = h + (H-h)//2
    print("A tippem: ", tipp)
    lepes += 1
    valasz = ' '
    while (valasz[0] not in "<>="):
        valasz = input("A gondolt szám <, > vagy = ezzel? ")
    _____:
        H = tipp          # ez a sor akkor hajtódjon végre, ha valasz[0] értéke '<'
    _____:
        h = tipp          # ez a sor akkor hajtódjon végre, ha valasz[0] értéke '>'
    _____:
        print(lepes, "lépésből kitaláltam!")          # egyébként ez fusson
        _____          # és utána egyből lépünk ki a ciklusból
    _____:
        print("Hmm... Biztosan nem csaltál?")          # ez a while ciklus else ága
```





A for ciklus

- egy bejárható halmaz elemeit a for ciklus segítségével járhatjuk be
- bejárható halmazok lehetnek pl. számsorozatok vagy karakterláncok (string-ek)
- számsorozatok a `range()` függvény segítségével képezhetünk
 - pl: `range(1, 100) → {1, 2, 3, ..., 99}`
 - vigyázat: a range felső határa nincs benne a halmazban!





Próbáld ki!

```
for n in range(1, 10):
    print(n, end=', ')
```

→ 1, 2, 3, 4, 5, 6, 7, 8, 9,

```
for n in range(1, 10, 2):
    print(n, end=', ')
```

→ 1, 3, 5, 7, 9,

```
for n in range(10, 1, -1):
    print(n, end=', ')
```

→ 10, 9, 8, 7, 6, 5, 4, 3, 2,

```
for n in range(10):
    print(n, end=', ')
```

→ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,



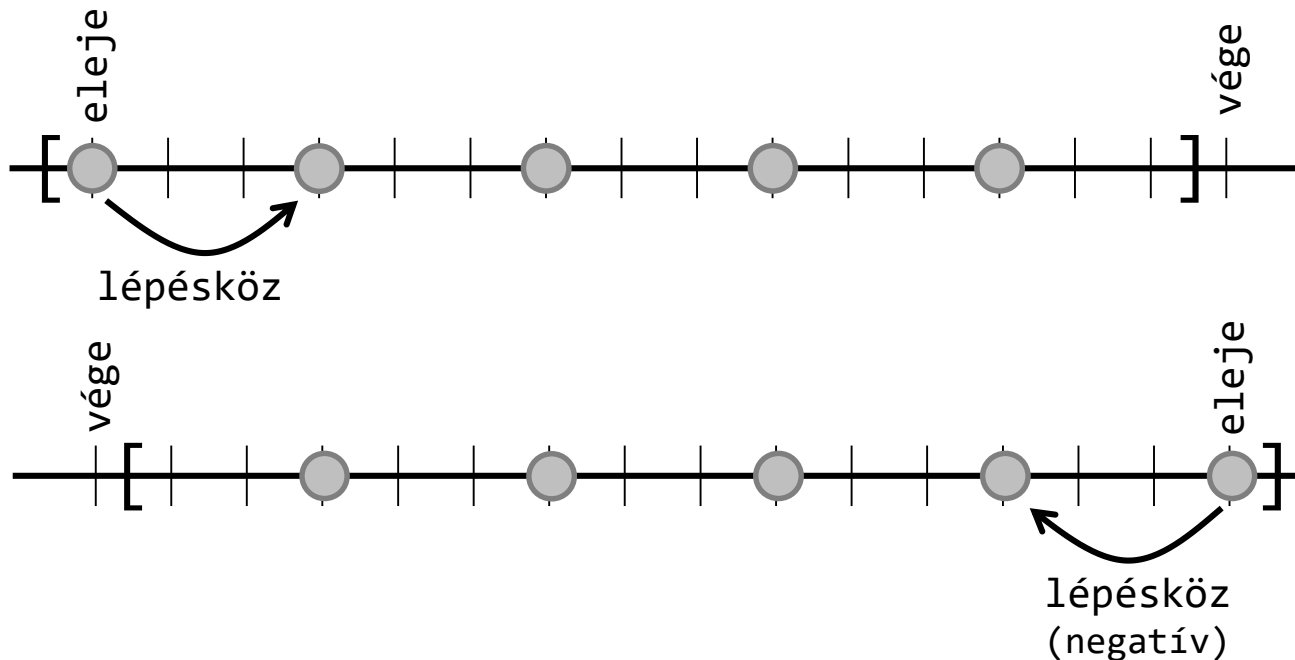


A tapasztalatok összegzése

ha nincs megadva,
akkor 0

ha nincs megadva,
akkor 1 (vagy -1)

```
range(eleje, vége, lépésköz)
```





Hasonlítsd össze!

```
n = 7
while (n <= 70):
    print(n)
    n = n + 7
```

```
for n in range(7, 71, 7):
    print(n)
```

```
n = 7
while (n <= 70):
    if (n == 7 or n == 70):
        print(n, "*")
    else:
        print(n)
    n = n + 7
```

```
for n in range(7, 71, 7):
    if (n == 7 or n == 70):
        print(n, "*")
    else:
        print(n)
```





Írd át for ciklusra!

```
from math import sqrt
szam = int(input("Írj be egy pozitív egész számot: "))
if (szam == 1):
    print("Definíció szerint nem prím.")
else:
    prim = True
    n = 2
    while (n <= sqrt(szam)):
        if (szam%n == 0):
            prim = False
            break
        n = n + 1
    if (prim):
        print("Prím.")
    else:
        print("Nem prím.")
```





Próbáld ki!

```
for c in "Szentendre":
    print(c, end='--')
```

→ S--z--e--n--t--e--n--d--r--e--

```
for c in "Szentendre":
    print(c, end='')
    if (c == 'e'):
        print('*', end='')
```

→ Sze*nte*ndre*

```
for c in "Szentendre":
    if (c != 'e'):
        print(c, end='')
```

→ Szntndr





Tufudsz íígy befeszéfnifi?

Írj egy fordító programot!

Írj egy olyan programot, ami bekér egy szöveget a felhasználótól, majd átfordítja azt egy olyan szövegre, melyben a magánhangzók a következő módon vannak kiegészítve:

a → afa

e → efe

i → ifi

...

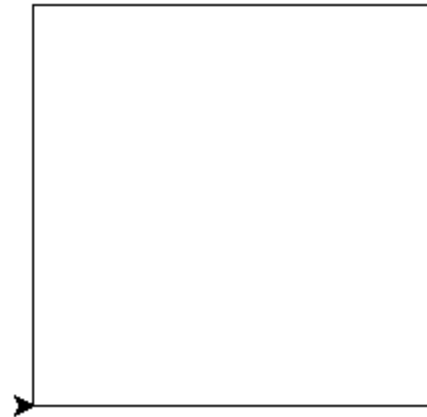
Például: helló → hefellófó





Próbáld ki!

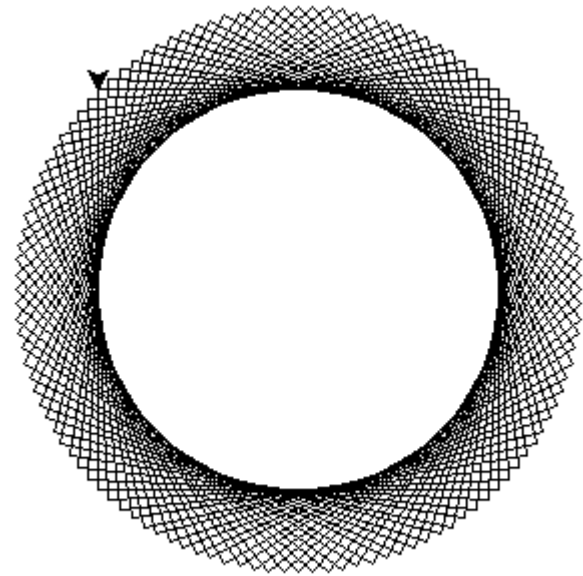
```
from turtle import *  
  
speed(3)  
  
for i in range(4):  
    forward(200)  
    left(90)  
  
done()
```





Próbáld ki!

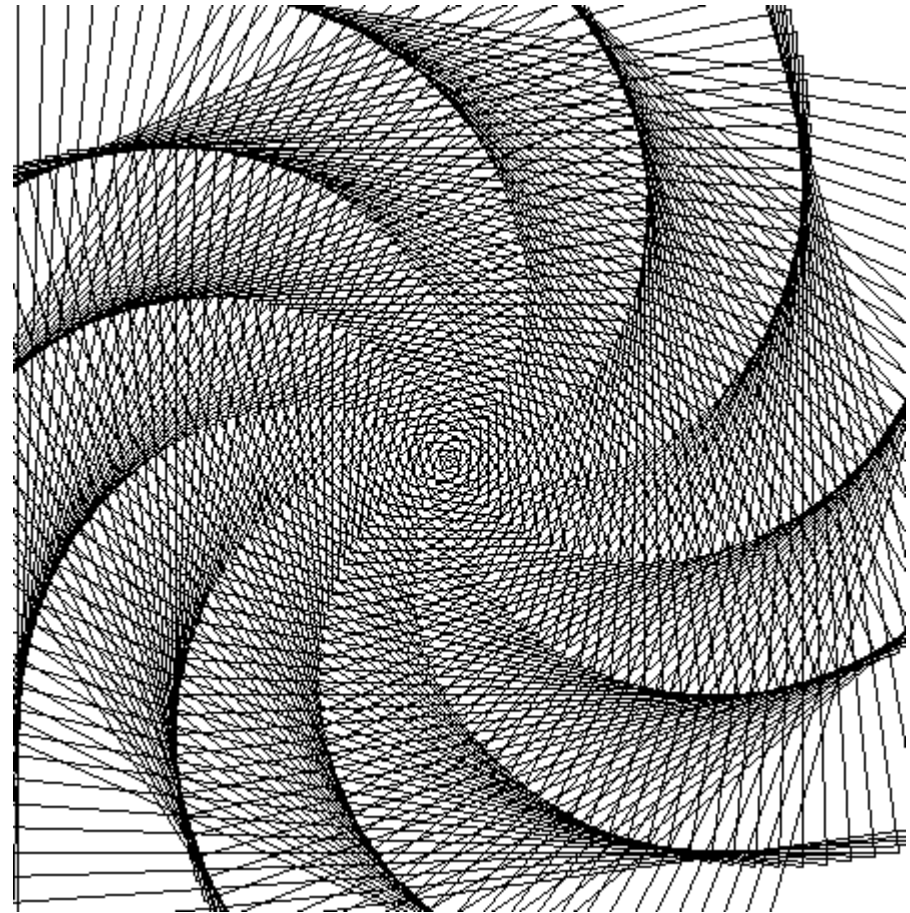
```
from turtle import *
from math import sqrt
speed(5)
m = sqrt(2)*200/2
d = 3
for a in range(30):
    for i in range(4):
        forward(200)
        left(90)
    penup()
    left(45)
    forward(m)
    left(180-d)
    forward(m)
    right(180+45)
    pendown()
done()
```





Próbáld ki!

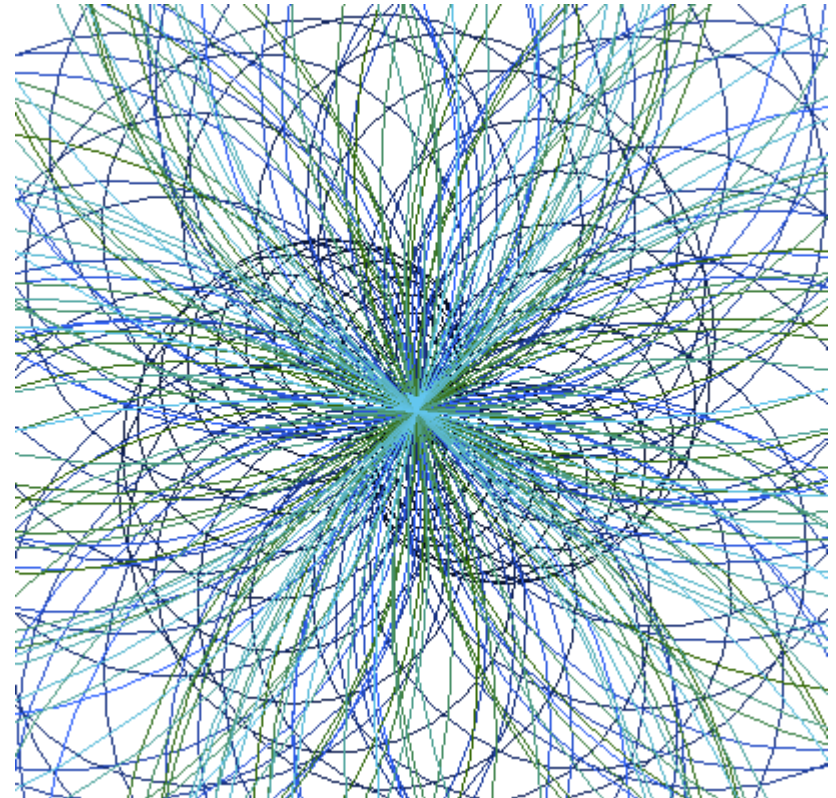
```
from turtle import *  
  
speed(0)  
  
for i in range(500):  
    forward(i)  
    left(262)  
  
done()
```





Próbáld ki!

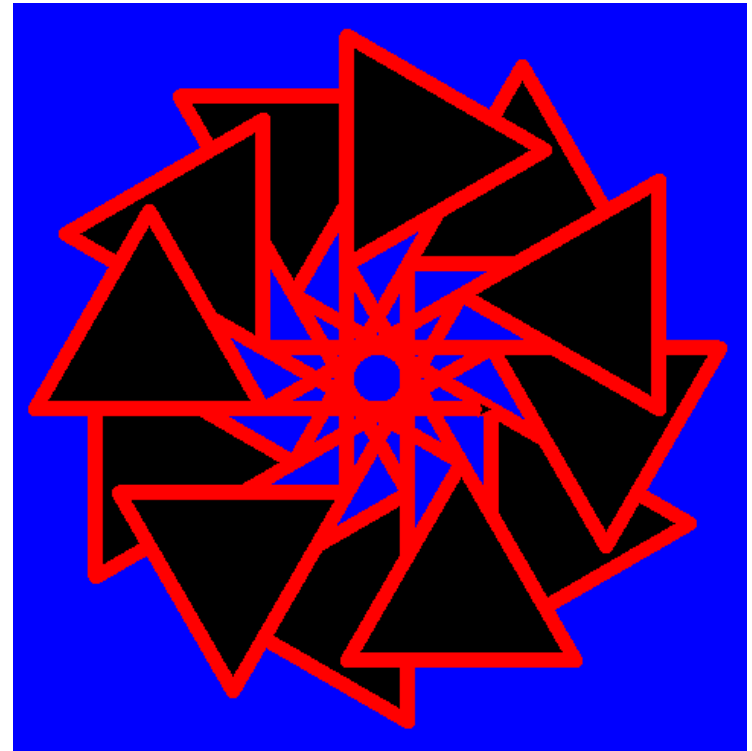
```
from turtle import *  
  
colormode(255)  
speed(0)  
  
for i in range(100):  
    circle(5*i)  
    circle(-5*i)  
    left(i)  
    color(i, 2*i%256, 5*i%256)  
  
done()
```





Próbáld ki!

```
from turtle import *  
  
bgcolor("blue")  
pencolor("red")  
pensize(10)  
for i in range(12):  
    begin_fill()  
    right(90)  
    forward(150)  
    left(120)  
    forward(150)  
    left(120)  
    forward(150)  
    end_fill()  
    forward(150)  
  
done()
```





Próbáld ki!

```
from turtle import *
from random import randint

speed(8)
setup(500, 500, 0, 0)
bgcolor("orange")

for i in range(100):
    x = randint(-250, 250)
    y = randint(-250, 250)
    goto(x, y)
    if (x > 50) and (x < 200) and (y > 50) and (y < 200):
        dot(20, "blue")
    else:
        dot(20, "green")
done()
```





Turtle modul

- teknőc parancsok elérése:

```
from turtle import *
```

- háttérszín, tollszín, és tollméret:

```
bgcolor("..."), pencolor("..."), pensize(...)
```

- toll letétele és toll felemelése:

```
pendown(), penup()
```

- mozgás előre, fordulás jobbra és balra:

```
forward(...), right(...), left(...)
```





Turtle modul

- kör és pont rajzolás:
`circle(...), dot(...)`
- terület beszínezése:
`fillcolor("...")`
`begin_fill();; end_fill()`
- teknőc sebessége:
`speed(...)` (`speed(0)` a leggyorsabb)
- rajzolás befejezése:
`done()`





Összefoglalás

- többszörös elágazás (`if-elif-else` szerkezet)
- a `while` ciklus kiegészítései (`break`, `continue`, `else`)
- a `for` ciklus és a `range()` függvény
- teknőc grafika a `turtle` modullal
 - példák a `for` ciklus alkalmazására
 - alapvető parancsok





EU zászló

Írj programot!

Írj programot, ami kirajzolja az EU zászlóját!





Készítette:
Buttyán Levente
Levente.Buttyan@gmail.com
CoderDojo Szentendre
2017