

Begiztak

Hasieran esan bezala, ordenagailuak oso tentelak dira. Esaten zaiena egiten dute. Baina, aldiz, gaitasun izugarriak eta pazientzia amaigabea dauzkate. Ez zaie batere inporta behar adina aldiz edozer gauza egitea.

Ordenagailu baten zeregin ohikoenetako bat agindu bat errepikatzea da. Hori begizta egituren bidez lor daitekeen zerbait da. Begizta ekintza errepikatu bat da. Oro har, begizta batek baldintza bat betetzen du exekutatzeko: baldintza horiek betetzen badira, orduan begizta horrek dituen aginduak exekutatu dira.

Begiztak errusiar mendiak bezalakoak dira, non zirkuitu itxi batean buelta batzuk ematen dituzun. Jarraian, begizta mota desberdinak ikusiko ditugu.

while begizta

Baldintza bat betetzen den bitartean, "while" begizta exekutatu egiten da. Egitura oso sinplea du:

```
while *baldintza*:  
    *eragiketak*
```

Adibidez, begizta bat exekutatu dugu aldagai baten balioa 0 baino handiagoa den bitartean.

```
kontagailua = 4  
while kontagailua > 0:  
    print ("Begiztaren barruan nago")  
    kontagailua = kontagailua - 1
```

Pantailako emaitza hau izango litzateke:

```
Begiztaren barruan nago  
Begiztaren barruan nago  
Begiztaren barruan nago  
Begiztaren barruan nago
```

Oharra: Kontuz! Konturatu al zara begiztaren barruan kontagailuari balio bat kentzen ari garela? Kontuz ez bagabiltza eta hori egitea ahazten badugu, kontagailuaren balioa ez litzateke inoiz aldatuko eta begizta infinitua sortuko genuke. Programa ez litzateke inoiz amaituko eta betiko trabatuta geratuko litzateke!

2.0 Ariketa

Sortu 0n hasitako kontagailu aldagai bat definituko duen programa. Ondoren, idatzi while begizta bat. Kontagailua 5 baino txikiagoa den bitartean, erakutsi mezu hau: "Begizta barruan nago".

```
kontagailua = 0
while kontagailua < 5:
    print("Begizta barruan nago")
    kontagailua = kontagailua + 1
```

Emitza:

```
Begizta barruan nago
Begizta barruan nago
Begizta barruan nago
Begizta barruan nago
Begizta barruan nago
```

Ikus dezagun beste adibide bat. Hurrengo programak datu bat eskatzen dio erabiltzaileari begizta batean. Programa ez da begiztatik aterako erabiltzaileak datu hutsa sartzen duen bitartean:

```
izena = ""
while izena == "":
    izena = input("Nola duzu izena?")

print("Kaixo", izena)
```

2.1 Ariketa

Idatzi erabiltzaileari bi zenbaki eskatzen dizkion programa bat. Lehenak bigarrena baino txikiagoa izan behar du. Begiztak bi zenbaki horien arteko tartean dauden zenbakiak erakutsi behar ditu.

```
min = input("Sartu minimo bat: ")
min = int(min)

max = input("Sartu maximo bat: ")
max = int(max)

while min < max:
    print(min)
    min = min + 1
```

Emitza:

```
Sartu minimo bat: 3
Sartu maximo bat: 8
3
4
5
```

```
6  
7
```

For begizta

for begizta ekintza bat errepikatzeko ere erabiltzen dugu. Baldintza bat baino gehiago, kontagailu moduko bat erabiltzen du:

```
for *aldagaia* in *tarte*:  
    *eragiketak*
```

Adibidez, honako begizta honek **"Kaixo"** mezua 4 aldiz erakutsiko du:

```
for i in range (4):  
    print ("Kaixo")
```

Hona hemen emaitza:

```
Kaixo  
Kaixo  
Kaixo  
Kaixo
```

For begiztetan oso interesgarria den zerbait **i** aldagaiak dira, nahi dugun izena izan lezakeenak, begiztaren buelta bakoitzari dagokion balioa izango duena. Hobeto ikusteko, nahikoa da aurreko programa pixka bat aldatzea:

```
for i in range (4):  
    print ("Kaixo", i)
```

Eta emaitzan ikusiko dugu nola aldatzen den **i** hori:

```
Kaixo 0  
Kaixo 1  
Kaixo 2  
Kaixo 3
```

Hori baliagarria izan daiteke programa askotan.

2.2 Ariketa

Idatzi erabiltzaileari zenbaki bat eskatzen dion programa bat, 0tik 10era biderkatzeko taula erakusten duena.

```
balioa = input("Sartu zenbaki bat: ")
balioa = int(balioa)

for i in range(11):
    print(balioa,"x",i,"=", (balioa*i))
    # Beste Aukerak:
    # print("%d x %d = %d" % (balioa, i, balioa * i))
```

Emitza:

```
Sartu zenbaki bat: 3
3 x 0 = 0
3 x 1 = 3
3 x 2 = 6
...etab.
```

2.3 Ariketa

Idatzi erabiltzaileari zenbaki bat eskatzen dion programa bat. 0 edo txikiagoa bada, zerbait handiagoa sartzeko adierazi behar duzu, eta bestela, "Python ondo dago!" mezua erakutsi behar duzu pantailan, zenbakiak adierazten duen adina aldiz:

```
zenbakia = input("Sartu zenbaki bat: ")
zenbakia = int(zenbakia)

if zenbakia <= 0:
    print("0 baino handiagoa den zenbaki bat sartu behar duzu")
else:
    for i in range(0, zenbakia):
        print("Python ondo dago!")
```

Emitza:

```
Sartu zenbaki bat: 3
Python ondo dago!
Python ondo dago!
Python ondo dago!
```

Tartea aldatu

Lehenetsita, `range (4)` 0tik 3ra bitarteko zerrenda itzultzen ari da, hau da: 0, 1, 2, 3. Lau elementu dira guztira eta, beraz, begiztak lau buelta emango ditu.

Jakina, edozein maila mota sor daiteke. Ezer adierazten ez bada, tartea 0an hasten da. Baina bi zenbakiren arteko tartea adieraz daiteke:

```
range (0, 4) # 0, 1, 2, 3
range (2, 6) # 2, 3, 4, 5
```

Adibidez:

```
for i in range (5, 9):
    print ("Kaixo", i)
```

Hona hemen emaitza:

```
Kaixo 5
Kaixo 6
Kaixo 7
Kaixo 8
```

Hirugarren parametro bat ere adieraz daiteke, balio batetik bestera nola egiten den jauzi adierazteko. Adibidez, 1tik 11ra, salto bikoitza emanez:

```
range (1, 11, 2) # 1, 3, 5, 7, 9
```

Hurrengo adibidean, begizta zenbaki bikoitiekkin egingo litzateke.

```
for i in range (0, 6, 2):
    print ("Kaixo", i)
```

Hona hemen emaitza:

```
Kaixo 0
Kaixo 2
Kaixo 4
```

Atzerantz

Begizta atzerantz ere ibil daiteke, jauzi negatiboa eginez:

```
print ("Atzerako kontaketa hasten:")  
for i in range (5, 0, -1):  
    print (i)
```

Hona hemen emaitza:

```
5  
4  
3  
2  
1
```

2.4 Ariketa

Idatzi erabiltzaileari zenbaki bat eskatzen dion programa bat. Ondoren, erabiltzaileak sartu duen zenbakira arteko zenbaki bikoiti guztiak erakutsi behar ditu, 0tik hasi eta Erabili **for** begizta eta egin salto binaka.

```
balioa = input("Sartu zenbaki bat: ")  
balioa = int(balioa)  
  
for i in range(0, balioa, 2):  
    print(i)
```

Emitza:

```
Sartu zenbaki bat: 5  
0  
2  
4
```

Zerrenden gaineko begiztak

Begizta bereziki erabilgarriak dira zerrenda bateko elementu guztiak aztertu nahi ditugunean, erakusteko, prozesatzeko edo dena delakoa. Forma oso erraza da:

```
objektuak = ["izarra", "perretxikoa", "lorea"]  
for objektua in objektuak:  
    print (objektua)
```

Begiztaren buelta bakoitzean, **objektu** aldagaiak balio bat hartuko du **objektuak** izeneko zerrendatik, eta, beraz, emaitza hau izango da:

```
izarra
perretxikoa
lorea
```

Begiztatik irteten

Batzutan, baliteke begizta batetik irten eta beste ezer prozesatzen ez jarraitzea. Demagun zerrenda baten barruan izen bat bilatzeko programa bat dugula:

```
izenak = ["Mia", "Jon", "Arya", "Ane", "Bug", "Ada", "Lisp"]
for izena in izenak:
    if izena == "Ane":
        print("Ane zerrendan dago")
```

Programa honek arazo bat dauka: nahiz eta **Ane** aurkitu, begiztak zerrendaren amaierara arte jarraituko du. Zerrenda hori oso handia bada, gure programa ez da eraginkorra izango! Hau da, denbora galtzen ibiliko da. Hasieran esan bezala, ordenagailuak oso tentelak dira. Gelditzeko esaten ez badiegu, aurrera jarraituko dute.

Zorionez, begiztetan **break** agindua erabil dezakegu, begizta bat-batean bukatzea lortuko duena:

```
izenak = ["Mia", "Jon", "Arya", "Ane", "Bug", "Ada", "Lisp"]
for izena in izenak:
    if izena == "Ane":
        print("Ane zerrendan dago")
        break
```

2.5 Ariketa

Idatzi 3 zenbakidun zerrenda bat definitzen duen programa bat. Ondoren, sortu **for** begizta bat, zerrenda zeharkatu eta zenbaki bakoitza bezain beste aldiz pantailatik **"Python"** hitza ateratzen duena.

```
errepikatu = [3, 6, 2]

for aldiz in errepikatu:
    for i in range(aldiz):
        print("Python")
```

Emitza:

```
Python
Python
Python
```

```
Python
```

```
...
```

Noiz erabili **while** ala **for**?

Biekin gauza bera egin zenezakeen arren, bakoitzak erabilera berezia du.

for begizta buelta kopuru zehatza exekutatu nahi denean erabiltzen da, ez gehiago, ez gutxiago. Edo, aurrerago ikusiko dugun bezala, datu-egiturak hasieratik amaierarainoko zerrendek kudeatu nahi direnean.

while begiztak ordeztu, baldintzak oso zehatzak ez direnean erabil daitezke. Adibidez, erabiltzaileak datu bat sartzea nahi badugu, begizta batean sartu dezakegu. Begizta ez da amaituko erabiltzaileak datu on bat sartu arte (hori litzateke baldintza).

Proposatutako ariketak

2.0 Ariketa

Idatzi programa bat **while** begizta batekin, erabiltzaileari izen bat eskatzeko (adibidez, "**Ada**") eta "**Kaixo Ada**" izen hori agurtzeko. "**irten**" testua sartuz gero, begizta amaitu egin behar da.

```
izena = ""

while izena != "irten":
    izena = input("Sartu izen bat: ")
    print("Kaixo", izena)

print("Amaiera.")
```

Emaita:

```
Sartu izen bat: Ada
Kaixo Ada
Sartu izen bat: Neko
Kaixo Neko
Sartu izen bat: irten
Kaixo irten
Amaiera.
```

2.1 Ariketa

Idatzi programa bat, erabiltzaileari zenbaki bat eskatzen diona eta zenbakia **0** ez den bitartean amaitzen ez duen **while** begizta duena. Zenbakia sartu ondoren, agur bat erakutsi behar da zenbakiak adierazten duen adina aldiz. Zenbakia **0** baino txikiagoa bada, begizta **break** batekin amaitu behar da;


```
balioa = ""

while balioa != 0:
    balioa = input("Sartu zenbaki bat: ")
    balioa = int(balioa)
    if balioa < 0:
        break

    for i in range(balioa):
        print("Kaixo")
```

Emitza:

```
Sartu zenbaki bat: 3
Kaixo
Kaixo
Kaixo
Sartu zenbaki bat: -1
```

2.2 Ariketa

Sortu erabiltzaileari balio oso bat eskatzen dion programa bat, egiaztatu 0 baino handiagoa dela eta, gainera, bikoitia dela. Hala bada, bistaratu pantailan * karakterea zenbakiaren balioa adina aldiz. Erabili `print ("*")`. Adibidez, 8 sartuz gero, hau erakutsiko du:

```
*****
```

Sartutako balioak baldintzak betetzen ez baditu, erabiltzailea ohartarazteko mezu bat erakutsi behar diozu eta programa amaitu.

```
zenbakia = input("Sartu zenbaki bat: ")
zenbakia = int(zenbakia)

if zenbakia <= 0 or zenbakia % 2 != 0:
    print("0 baino handiagoa den zenbaki bikoitia sartu behar duzu")
else:
    izarrak = ""
    while zenbakia > 0:
        izarrak = izarrak + "*"
        zenbakia = zenbakia - 1

    print(izarrak)
```

Emitza:

```
Sartu zenbaki bat: 6
*****
```

2.3 Ariketa

Sortu aurrekoaren antzeko programa bat, zenbaki bikoiti eta positiboak bakarrik onartu behar dituen. Sartutako zenbakiarekin `*` testua pantailan agertu behar da, `*`rekin amaituta. Adibidez:

8 zenbakia sartzen bada, hauxe erakutsiko da:

```
*-*-*-*
```

Adibidez, 4 zenbakia sartzen bada:

```
*-*-*
```

Programa:

```
zenbakia = input("Sartu zenbaki bat: ")
zenbakia = int(zenbakia)

if zenbakia <= 0 or zenbakia % 2 != 0:
    print("0 baino handiagoa den zenbaki bikoitia sartu behar duzu")
else:
    sekuentzia = ""
    zenbakia = zenbakia / 2
    while zenbakia > 0:
        sekuentzia = sekuentzia + "*-"
        zenbakia = zenbakia - 1

    sekuentzia = sekuentzia + "*"

    print(sekuentzia)
```

Emitza:

```
Sartu zenbaki bat: 10
*-*-*-*-*
```

2.4 Ariketa

Erabiltzaileari zenbaki oso bat eskatzen dion programa bat sortu, eta balio hori erabiliz, lauki bat marraztu behar du pantailan * karakterea erabiliz.

Adibidez, 4 sartuz gero, hau erakutsiko da:

```
****
****
****
****
```

```
zenbakia = input("Sartu zenbaki bat: ")
zenbakia = int(zenbakia)

if zenbakia <= 0:
    print("0 baino handiagoa den zenbaki bat sartu behar duzu")
else:
    izarrak = "\n"
    for i in range(zenbakia):
        for j in range(zenbakia):
            izarrak = izarrak + "*"

        izarrak = izarrak + "\n"

    print(izarrak)
```

Emitza:

```
Sartu zenbaki bat: 2
**
**
```

2.5 Ariketa

Sortu programa bat erabiltzaileari zenbaki oso bat eskatu eta haren faktoria kalkulatzeko. Adibidez, 5 zenbakiaren faktoria $5 \times 4 \times 3 \times 2 \times 1 = 120$ izango litzateke.

```
zenbakia = input("Sartu zenbaki bat: ")
zenbakia = int(zenbakia)

if zenbakia <= 0:
    print("0 baino handiagoa den zenbaki bat sartu behar duzu")
else:
    faktoriala = zenbakia
    while zenbakia > 1:
        zenbakia = zenbakia - 1
        faktoriala = faktoriala * zenbakia
```

```
print("Emitza: ", faktoriala)
```

Emitza:

```
Sartu zenbaki bat: 4
Emitza: 24
```

2.6 Ariketa

Sortu erabiltzaileari zenbaki oso bat eskatzen dion programa bat eta egiaztatu zenbaki hori lehena (primoa) den ala ez, hau da, bakarrik bere buruarekin edo 1 egatik zatigarria dena. Adibidez: 2, 3, 5, 7, 11, 13, 17,...

```
zenbakia = input("Sartu zenbaki bat: ")
zenbakia = int(zenbakia)

if zenbakia <= 0:
    print("0 baino handiagoa den zenbakia sartu behar duzu")
else:
    zatigarria = False
    hasierakoa = zenbakia
    zenbakia = zenbakia - 1

    while zenbakia > 1 and not zatigarria:
        if hasierakoa % zenbakia == 0:
            zatigarria = True

        zenbakia = zenbakia - 1

    if not zatigarria:
        print(hasierakoa, " zenbakia lehena da.")
    else:
        print(hasierakoa, " lehen ez da.")
```

Emitza:

```
Sartu zenbaki bat: 5
5 zenbakia lehena da.
```

2.7 Ariketa

Sortu 0tik 10era biderketa taula guztiak erakusten dituen programa bat.

```
for i in range(11):
    for j in range(11):
        print(i,"x",j,"=",i*j)

# Honek ere berdin egingo du
for i in range(11):
    for j in range(11):
        print(" %d x %d = %d" % (i, j, i*j))
```