

Ku celcelinta (Iteration) qaababka xogta (data structures)

Computing

Unit Barnaamijyada Python ee taxanaha xogta (sequences of data)



Natiijo

Waxaan isticmaali karaa ku celcelinta (iteration) si aan si joogto ah wax ugu daro liisaska (lists) iyo inaan hubiyo waxa ku jira xarfo (string).

Erayada muhiimka ah

ku celcelin (iteration)

habka lagu celceliyo taxane tilmaamo (instructions) gudaha loop-ka barnaamijka

xarfo (string)

taxane xarfo ah, tusaale: xarfaha, tirooyinka, meelaha bannaan, iyo astaamo kale

mutable (la beddeli karo)

ahaan kara mid la beddelo

immutable (aan la beddeli karin)

aan awoodin in la beddelo

Ku celcelinta (Iteration) qaababka xogta (data structures)



Isticmaal ku celcelinta (iteration) si joogto ah wax ugu darto liiska (list)



Sharax sida xarfaha (strings) uga duwan yihiin liisaska (lists)



Ku celceli xarfaha (string) si aad u barbardhigto xarfaha

Barnaamijkani wuxuu qaataa magaca qalabka muusikada wuxuuna ku darayaa liis la yiraahdo band.

```
1 print("Let's form a band")
2 band = []
3 print("Pick an instrument")
4 instrument = input()
5 band.append(instrument)
6 print(band)
```

Falkani wuxuu u baahan yahay in si joogto ah loo sameeyo, si loogu daro dhammaan qalabka loo baahan yahay si loo sameeyo koox.



```
1 print("Let's form a band")
2 band = []
3 print("Pick an instrument")
4 instrument = input()
5 band.append(instrument)
6 print("Pick an instrument")
7 instrument = input()
8 band.append(instrument)
9 print("Pick an instrument")
10 instrument = input()
11 band.append(instrument)
12 print(band)
```

Tani uma eka qaabka ugu hufan (efficient) ee loo qaban karo!



Saciid

Maxaad samayn kartaa si aad u soo celceliso (repeatedly) ficillada (actions) barnaamijkan?



```
1 print("Let's form a band")
2 band = []
3 print("Pick an instrument")
4 instrument = input()
5 band.append(instrument)
6 print("Pick an instrument")
7 instrument = input()
8 band.append(instrument)
9 print("Pick an instrument")
10 instrument = input()
11 band.append(instrument)
12 print(band)
```

Waxaad isticmaali kartaa iteration (ku celcelin)!



Nuur

Haa, while loop (wareeg) wuxuu ku celceliyaa ficol ilaa shuruud la buuxiyo.



Qaab dhismeedkee barnaamijka ayaa loo isticmaalaa si loo ku celceliyo ficilada barnaamijka?

a

selection (xulasho)

b

sequence (isku xigxig)

c

iteration (ku celcelin)



Barnaamijka hadda waa la beddelay si loo isticmaalo iteration (ku celcelin).

```
1 print("Let's form a band")
2 band = []
3 while len(band)<3:
4     print("Dooro qalab muusik")
5     instrument = input()
6     band.append(instrument)
7     print(band)
```

Habka len() list waxaa loo isticmaalaa in lagu hubiyo dhererka liiska.

Isticmaal iteration si aad ugu sii darto (append) shay liiska



```
1 print("Aynu samayno koox")
2 band = []
3 while len(band)<3:
4     print("Dooro qalab muusik")
5     instrument = input()
6     band.append(instrument)
7     print(band)
```

Soo saarista qoraalka (Text output)

```
Aynu samayno koox
Dooro qalab muusik
Drum
Dooro qalab muusik
Guitar
Dooro qalab muusik
Piano
['Durbaan', 'Gitaar', 'Piano']
```

Habka waa la soo celceliyaa inta dhererka liisku aanu gaarin 3.

Kooxdu waxay caddeysay inay jiri karo ciyaaryahan gitaar ah.

```
1 print("Aan samayno koox")
2 band = []
3 while "Gitaar" not in band:
4     print("Dooro qalab")
5     instrument = input()
6     band.append(instrument)
7     print(band)
```

Hawlgalka 'not in' waxaa loo isticmaalaa in lagu hubiyo in shayga 'Gitaar' uusan ku jirin liiska.



```
1 print("Aan samayno koox")
2 band = []
3 while "Gitaar" not in band:
4     print("Dooro qalab")
5     instrument = input()
6     band.append(instrument)
7     print(band)
```

Soo-saar qoraal (Text output)

```
Aan sameyno koox (band)
Dooro qalab (instrument)
Durbaan (Drum)
Dooro qalab (instrument)
Violin (Violin)
Dooro qalab (instrument)
Piano (Piano)
Dooro qalab (instrument)
Gitaar (Guitar)
['Durbaan (Drum)', 'Violin (Violin)', 'Piano (Piano)', 'Gitaar (Guitar)']
```

Habka waa la ku celceliyaa inta shayga 'Gitaar' uusan ku jirin liiska.

Run mise been?

Loop-ka 'while' (while loop) wuxuu socon doonaa tiro go'an oo jeer (set number of times).



Run



Been



Loop-ka 'while' (while loop) wuxuu sameeyaa xakamaynta shuruudda (condition-controlled iteration), taasoo la micno ah inuu socon doono ilaa shuruud (condition) la buuxiyo.

- 1) Fur barnaamijka bilowga (starter program) oak.link/city-hopping
- 2) Ku dar sadarradan koodka (lines of code) si barnaamijkaagu u doorto si random ah (randomly) shan magaalo (cities) oo ka mid ah liiska magaalooyinka (cities list) oo ugu soo cel-celiyo (appends) liiska safarka (trip list). Fiiri: Sadarrada koodka hadda kuma jiraan nidaamka saxda ah (correct order).

```
print("City hopping random planner")
print(f"Itinerary: {trip}")
trip = []
city = choice(cities)
trip.append(city)
halka inta len(trip) < 5:
```

- 3) Socodsii barnaamijka dhowr jeer si aad u hubiso inuu si random ah (randomly) u doorto shan magaalo (cities).

4) Waxaa laga yaabaa in aad ogaato in mararka qaarkood liiska shanta magaalo uu ka kooban yahay magaalo soo noqnoqota. Hagaaji barnaamijkaaga si aan magaalooyin duplicate ah loogu darin liiska safarka.

Tilmaan: Waxaad u baahan doontaa inaad isticmaasho habka `list.remove()`.

5) Waxaad rabtaa inaad booqato London. Hagaaji barnaamijkaaga si aad uga badin liiska shanta magaalo, habka doorashada magaalo random wuxuu sii socdaa ilaa London la doorto oo lagu daro liiska safarka.

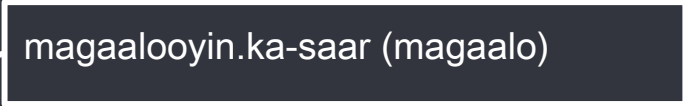
oak.link/city-hopping-solution

2)

```
1 from random import choice
2 cities=["Athens","Berlin","Dublin","London", "Madrid","Paris","Prague","Stockholm", "Vienna"]
3
4
5
6
7
8 print("City hopping random planner")
9 trip = []
10 halka inta len(trip) < 5:
11     city = choice(cities)
12     trip.append(city)
13     print(f"Itinerary: {trip}")
```

4)

```
1 from random import choice
2 cities=["Athens","Berlin","Dublin","London", "Madrid","Paris","Prague","Stockholm", "Vienna"]
3
4
5
6
7 print("City hopping random planner")
8 trip = []
9 halka inta len(trip) < 5:
10 magaalo = dooro (magaalooyin)
11 safar.ku-dar (magaalo)
12 magaalooyin.ka-saar (magaalo)
   daabac(f"Jadwalka Safarka: {safar}")
```



5)

```
1  laga soo xigo random (random) dooro (choice)
2  magaalooyin=["Athens","Berlin","Dublin","London", "Madrid","Paris","Prague","Stockholm", "Vienna"]
3
4
5
6
7  daabac("Qorshe-sameeye magaalo-boodbood oo random ah")
8  safar = []
9  halkii "London" kuma jiro safar:
10 magaalo = dooro (magaalooyin)
11 safar.ku-dar (magaalo)
12 magaalooyin.ka-saar (magaalo)
   daabac(f"Jadwalka Safarka: {safar}")
```

halkii "London" kuma jiro safar:

Qaabka casharka

Ku-celcelinta (iterating) qaab-dhismeedka xogta (data structures)



Isticmaal ku-celcelin (iteration) si aad wax ugu sii darto (append) liis (list)



Sharax sida xargaha (strings) uga duwan yihiin liisaska (lists)



Ku-celceli (iterate) xarig (string) si aad isbarbardhigto xarfaha (characters)

Sharax sida xargaha (strings) uga duwan yihiin liisaska (lists)



Xarig (string) waa nooc xog (data type). Xarig (string) waa urur xarfo ah oo ka kooban kara meelo bannaan, calaamado (punctuation), tirooyin (numeric digits) iyo calaamado kale sida \$, & iyo £.

Python-ka, waxaan u matallaynaa string (string) iyadoo la dhigo " " agagaarka xarfaha. Tusaale, name = "Laura" ama password = "Password1%".

Fiiro gaar ah: Haddii string (string) uu ka kooban yahay tirooyin, kuwaas waxaa loola dhaqmi doonaa sida xarfaha qoraalka, ma aha tirooyin.

Sharax sida strings (strings) ay uga duwan yihiin lists (lists)



Run mise been?

String (string) ma qaadi karo qiimayaal (values) tirooyin ah.



Run



Been



String (string) wuxuu qaadi karaa qiimayaal (values) tirooyin ah, laakiin kuwaas waxaa loola dhaqmi doonaa sida xarfaha qoraalka, ma aha tirooyin.

Qaar ka mid ah hawlo (functions) Python ah ayaa loo isticmaali karaa labadaba strings (strings) iyo lists (lists).

Aan eegno hawsha (function) len():

List (list)

List-ka (list), hawsha (function) len waxay soo celisaa dhererka (tirada shayga) ee list-ka.

```
numbers = [16, 76, 11]
length = len(numbers)
```

String (string)

String-ka (string), hawsha (function) len waxay soo celisaa dhererka (tirada xarfaha) ee string-ka.

```
language = "Python"
length = len(language)
```

Hadda, aan eegno hawlgalka (operator) in:

List (list)

Hawlgalka (operator) in wuxuu hubinayaa in qiime (value) uu la mid yahay shay kasta oo ku jira list-ka.

Tibaaxaha (expressions) la sameeyo waxay qiimeyn doonaan True ama False.

```
numbers = [16, 76, 11]
found = 32 in numbers
```

String (string)

The **in** operator checks if a **string** is contained within another **string**.

Tibaaxaha la sameeyo waxay qiimeyn doonaan Run ama Been.

```
language = "Python"
contains = "ty" in language
contains = "on" in language
```

How about indexing?

List

An item in a list can be accessed through its index.

`list[index]`

```
numbers = [16, 76, 11]
```

```
item = numbers[1]
```

```
numbers[3] = 32
```

76

String

A character in a **string** can be accessed through its index.

`string[index]`

```
language = "Python"
```

```
character = language[1]
```

y

Sharax sida strings (xarfo) ay uga duwan yihiin lists (liisaska)



Saciid

Marka, ma waxaan ku cusboonaysiin karaa xarfaha ku jira string (xarfo) adigoo isticmaalaya index qiime (value)?

```
language = "Python"  
character = language[1]  
language[0] = "p"
```

Qoraal soo saar (Text output)

Khaladka Nooca: shayga 'str' ma taageero meelaynta shayga safka 3 ee main.py

Ma awoodid inaad u qoondeyso qiime (value) cusub xaraf gaar ah oo ku jira string (xarfo).

Lists (liisaska) waa mutable (la beddeli karo), taas oo macnaheedu yahay in waxa ku jira la beddeli karo, iyada oo aan loo baahnayn in la sameeyo list (liis) cusub.

Strings (xarfaha) waa immutable (aan la beddeli karin), sidaas darteed walxaha lama beddeli karo, lama kordhin karo, lamana saari karo.

Haddii aad rabto inaad cusboonaysiiso string (xarfo), waxaad u baahan tahay inaad dib u qoondeyso qiime (value) dhan oo uu haysto variable (doorsoome), taas oo qori doonta qiimihii (value) hore.

Koodhkee ayaad isticmaali kartaa si aad u hubiso inta dheer ee string (xarfo) “Oak National Academy”?

a `len(string)`



b `in`

c `string[index]`

Maxaan u helayaa jawaabta 20 ee ma aha 18?

`len(string)` waxay soo celin doontaa 20 sababtoo ah waxay tirisaa meelaha bannaan (spaces) sidoo kale.



Nuur

- 1) Sharax sida xadkhaha (strings) uga duwan yihiin liisaska (lists) adigoo isticmaalaya erayadan hoose.

Aan la beddeli karin (Immutable) La beddeli karo (Mutable)



- 1) Sharax sida xadkhaha (strings) uga duwan yihiin liisaska (lists) adigoo isticmaalaya erayadan hoose.

Inkasta oo hawlo badan oo isku mid ah lagu sameyn karo xadkhaha (strings) iyo liisaska (lists), waxay u dhaqmaan si ka duwan. Liisku waa la beddeli karaa (mutable), taas oo macneheedu yahay qiimaha (values) ku jira liiska waa la cusboonaysiin karaa. Xadhkuhu (string) waa aan la beddeli karin (immutable), taas oo macneheedu yahay xurufta (characters) gaarka ah lama cusboonaysiin karo, lama beddeli karo, lamana kordhin karo.

Qaabka casharka

Soo noqnoqoshada (Iterating) qaababka xogta (data structures)



Isticmaal soo noqnoqoshada (iteration) si aad si isdaba joog ah ugu darto shay (items) liis (list)



Sharax sida xadkhaha (strings) uga duwan yihiin liisaska (lists)



Soo noqnoqo (Iterate) xadhka (string) si aad isbarbardhigo xurufta (characters)



Sideen u isbarbardhigi karaa xurufta (characters) xadhka (string) ku jira?

Saciid

Xusuusnow, waxaad isticmaali kartaa hawl wadeennada isbarbardhigga (relational operators) si aad u isbarbardhigto qiimaha (values).



Soo noqnoqo (Iterate) xadhka (string) si aad isbarbardhigo xurufta (characters)



==	la mid ah (equal to)
!=	aan la mid ahayn (not equal to)
kuma jiro (not in)	hubiyaa haddii qiimaha (value) ku jiro xadhka (string)

Soo noqnoqo (Iterate) xadhka (string) si aad isbarbardhigo xurufta (characters)



Waa kuwan tusaalooyin adeegsanaya xarfaha name1 = "Layla" iyo name2 = "Nuur".

<code>name1[0] == name2[0]</code>	hubi xarafka koowaad ee name1 inuu la mid yahay xarafka koowaad ee name2	False
<code>len(name1) != len(name2)</code>	hubi dhererka name1 inuusan la mid ahayn dhererka name2	True
<code>name1[0] not in name2</code>	hubi xarafka koowaad ee name1 inuusan ka muuqan meel kasta oo name2 ah	True

Ku wareeg xarfo si aad u barbar dhigto xarfaha



Waa kuma koodka aad isticmaali lahayd haddii aad rabto inaad hubiso qiimaha (value) xarfka koowaad ee ku jira xarfka language = "Python"?

a

```
language["P"]
```

b

```
language[1]
```

c

```
language[0]
```



Xusuusnow, Python-ka, index-ku wuxuu ka bilaabmaa eber.

Ku wareeg xarfo si aad u barbar dhigto xarfaha



Waa kuma koodka aad isticmaali lahayd si aad u hubiso in xarfka koowaad ee name1 uusan la mid ahayn xarfka koowaad ee name2?

a `name1[0] == name2[0]`

b `name1[1] == name2[1]`

c `name1[0] != name2[0]`



Barnaamijyada, calaamadaha (flags) waxaa loo isticmaalaa in loo sheego barnaamijka in shuruud la buuxiyay iyo in kale.

Barnaamijkan, while True: waxaa loo isticmaalaa calaamad (flag) ah, barnaamijku wuxuu sii socon doonaa waligiis ilaa calaamadda la beddelo.

```
1 count = 1
2 while True:
3     print(count)
4     count = count + 1
```

Ciyaar (game) ayaa la sameeyay halkaas oo magaalo madaxda (capital city) si random ah looga soo xusho liis (list) ka dibna isticmaalaha (user) waxaa laga codsadaa inuu qiyaaso (guess) magaalo madaxda.

- 1) Fur barnaamijka (starter program) oak.link/guess-the-capital
- 2) Ku socodsii (Run) barnaamijka dhowr jeer si aad u aragto sida uu u shaqeeyo.
- 3) Barnaamijka hadda wuxuu u oggolaanayaa isticmaalaha (user) inuu qiyaaso (guess) magaalo madaxda hal mar. Ku dar loop (while loop) line 10 kaas oo socon doona inta calaamadda (flag) done = False.

Tilmaan (Hint): Waxaad u baahan doontaa inaad beddesho indentation-ka (indentation) code-ka ku jira loop-kaaga (while loop).

- 4) Ka beddel (Amend) code-ka si aad isticmaalaha (user) u siiso caawinaad (help).
 - a) Ku dar hadal (elif statement) si barnaamijku u muujiyo xarafka koowaad (first letter) ee magaalada (city) isticmaalaha (user), haddii xarafka koowaad ee magaalada uu ka duwan yahay xarafka koowaad ee qiyaasta (guess) isticmaalaha.
 - b) Ku dar hadal (elif statement) si barnaamijku u muujiyo dhererka (length) (tirada xuruufaha) ee magaalada (city) isticmaalaha (user), haddii dhererka magaalada uu ka duwan yahay dhererka qiyaasta (guess) isticmaalaha.
 - c) Ku dar hadal (elif statement) si barnaamijku u muujiyo xarafka labaad (second letter) ee magaalada (city) isticmaalaha (user), haddii xarafka labaad ee magaalada uusan ku jirin qiyaasta (guess) isticmaalaha.

Soo saarista qoraalka (Text output)

Qiyaas (Guess) magaalo madaxda (capital):

Athens

Xarafka koowaad (first letter) waa: S
Qiyaas (Guess) magaalo madaxda (capital):

Sofia

Waxay leedahay 9 xaraf (letters)
Qiyaas (Guess) magaalo madaxda (capital):

Stockholm

Waa kuu helay!

```
1 from random import choice
2
3 cities = ["Athens", "Berlin", "Brussels", "Copenhagen",
4 "Dublin", "London", "Madrid", "Paris",
5 "Prague", "Stockholm", "Vienna"]
6
7 city = choice(cities)
8 done = False
9
10 while done == False:
11     print("Guess the capital:")
12     guess = input()
13     if guess == city:
14         print("You've got it!")
15         done = True
16     elif guess == "":
17         print(f"It was {city}")
18         done = True
19     elif city[0] != guess[0]:
20         print(f"The first letter is: {city[0]}")
21     elif len(guess) != len(city):
22         print(f"Waxay leedahay {len(city)} xaraf")
23     elif city[1] kuma jiro guess:
24         print(f"Waxay ka kooban tahay xarafka {city[1]}")
25     else:
26         print("Mar kale isku day")
```

3) Barnaamijku hadda wuxuu u oggol yahay qofka isticmaala inuu hal mar qiyaaso magaalada caasimadda ah. Ku dar loop-ka 'while' line-ka 10 kaas oo shaqayn doona inta calaamadda 'done = False' ay tahay.

while done == False:

oak.link/guess-the-capital-solution

```
1 from random import choice
2
3 cities = ["Athens", "Berlin", "Brussels", "Copenhagen",
4 "Dublin", "London", "Madrid", "Paris",
5 "Prague", "Stockholm", "Vienna"]
6
7 city = choice(cities)
8 done = False
9
10 while done == False:
11     print("Qiyaas caasimadda:")
12     guess = input()
13     if guess == city:
14         print("Waa kuu sugan!")
15         done = True
16     elif guess == "":
17         print(f"It was {city}")
18         done = True
19     elif city[0] != guess[0]:
20         print(f"The first letter is: {city[0]}")
21     elif len(guess) != len(city):
22         print(f"It has {len(city)} letters")
23     elif city[1] not in guess:
24         print(f"It contains letter {city[1]}")
25     else:
26         print("Try again")
```

```
elif city[0] != guess[0]:
    print(f"The first letter is: {city[0]}")
```

4a) Ku dar hadal elif si barnaamijku u muujiyo xarafka koowaad ee magaalada isticmaalaha, haddii xarafka koowaad ee magaaladu ka duwan yahay xarafka koowaad ee qiyaasta isticmaalaha.

oak.link/guess-the-capital-solution

```
1 from random import choice
2
3 cities = ["Athens", "Berlin", "Brussels", "Copenhagen",
4 "Dublin", "London", "Madrid", "Paris",
5 "Prague", "Stockholm", "Vienna"]
6
7 city = choice(cities)
8 done = False
9
10 while done == False:
11     print("Guess the capital:")
12     guess = input()
13     if guess == city:
14         print("You've got it!")
15         done = True
16     elif guess == "":
17         print(f"It was {city}")
18         done = True
19     elif city[0] != guess[0]:
20         print(f"The first letter is: {city[0]}")
21     elif len(guess) != len(city):
22         print(f"It has {len(city)} letters")
23     elif city[1] not in guess:
24         print(f"It contains letter {city[1]}")
25     else:
26         print("Try again")
```

4b) Ku dar elif si barnaamijku u muujiyo dhererka (tirada xarfaha) ee city isticmaalaha, haddii dhererka city uu ka duwan yahay dhererka qiyaasta isticmaalaha.

```
elif len(guess) != len(city):
    print(f"It has {len(city)} letters")
```

oak.link/guess-the-capital-solution



```
1 from random import choice
2
3 cities = ["Athens", "Berlin", "Brussels", "Copenhagen",
4           "Dublin", "London", "Madrid", "Paris",
5           "Prague", "Stockholm", "Vienna"]
6
7 city = choice(cities)
8 done = False
9
10 while done == False:
11     print("Guess the capital:")
12     guess = input()
13     if guess == city:
14         print("You've got it!")
15         done = True
16     elif guess == "":
17         print(f"It was {city}")
18         done = True
19     elif city[0] != guess[0]:
20         print(f"The first letter is: {city[0]}")
21     elif len(guess) != len(city):
22         print(f"It has {len(city)} letters")
23     elif city[1] not in guess:
24         print(f"It contains letter {city[1]}")
25     else:
26         print("Try again")
```

4c) Ku dar hadal elif (elif statement) si barnaamijku u muujiyo xarafka labaad ee magaalada (city) isticmaalaha, haddii xarafka labaad ee magaalada uusan ku jirin qiyaasta (guess) isticmaalaha.

```
elif city[1] not in guess:
    print(f"Waxaa ku jira xarafka {city[1]}")
```

oak.link/guess-the-capital-solution

Ku celcelinta xaaladda (Condition-controlled iteration) waxaa lagu hirgeliyay iyadoo la isticmaalayo loop-ka while (while loops) ee Python.

Ku celcelinta (Iteration) waxaa loo isticmaali karaa in si isdaba joog ah loogu daro shay (items) liis (list).

Xarig (string) wuxuu la mid yahay liis (list) qaab ahaan laakiin waa mid aan la beddeli karin (immutable), marka curiyeyaasha (elements) lama beddeli karo, lama kordhin karo, lamana qaadi karo.