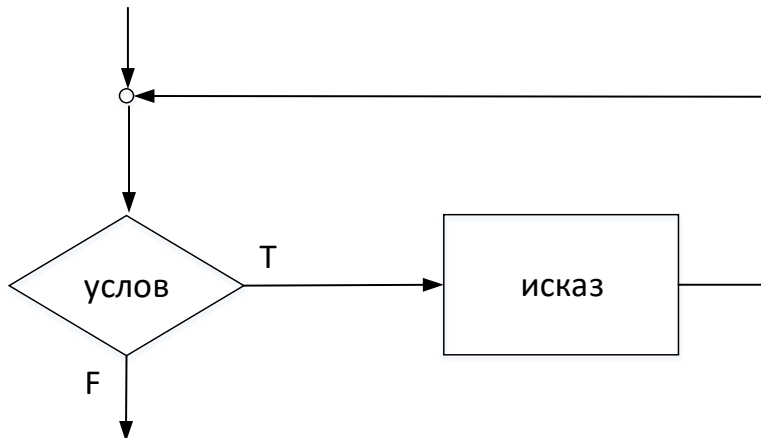


Петље у програмирању

Уместо да се пише исти код потребан број пута, логичније је написати део кода једном па програмирати да се понавља потребан број пута. Ово се назива програмерска петља или структура понављања.

У Пајтону постоје две врсте петље: `while` петља (петља контролисана условом) и `for` петља (петља контролисана бројачем).

While петља

Петље контролисане условом се понављају док је услов испуњен.

Назив `while` петља потиче од објашњења на енглеском језику који приказује како ради петља: `while a condition is true, do some task` (док је услов испуњен, уради неки задатак).

`While` петља има два дела: услов који се тестира и низ исказа (тело петље) који се понављају све док је услов тачан.

Из дијаграма се види да се после извршења исказа ток кода враћа на поновно испитивање услова петље.

Општи формат `while` петље:

```
while uslov:
    iskaz
```

Пример 01: Рад `while` петље

```
lep_dan = 'da'
while lep_dan == 'da':
    lep_dan = input("Da li je danas lep dan? ")
print("Vise nije.")
Da li je danas lep dan? da
Da li je danas lep dan? da
Da li je danas lep dan? da
Da li je danas lep dan? ne
Vise nije.
```

Из кода се види да линија `while lep_dan == 'da':` представља реченицу `while`, где је само `lep_dan == 'da'` услов петље који се проверава.

Ако је услов испуњен, извршиће се `lep_dan = input("Da li je danas lep dan? ")` које је тело петље (исказ) па се поново услов проверава.

Када услов више није испуњен, извршава се `print("Vise nije.")` исказ који није део `while` петље.

Види се да промењива `lep_dan` има иницијалну вредност "да". То је зато што ако не би имала баш ту вредност као иницијалну, програм никад не би испунио услов за улазак у петљу.

Једно извршење тела петље се назива циклус или итерација.

`While` петља је врста петље која се назива и петља са излазом на врху, а то значи да се увек прво провери тачност услова па се тек онда извршава тело петље.

Такође, пошто се не зна унапред у ком тренутку ће корисник унети другачију вредност за промењиву `lep_dan`, ова врста петље се назива и кориснички контролисана петља.

Пример 02: Корисничка контрола петље

```
print("Potrebno je voditi racuna o temperaturi u peci.")
print("Temperatura ne sme pasti ispod 50 celzijusovih stepeni.")
print("Ako se to desi odmah se mora povecati temperatura u peci.")
temperatura = float(input("Kolika je temperatura u peci? "))
while temperatura >= 50:
    print("Temperatura je u pogodna za odrzavanje procesa.")
    temperatura = float(input("Kolika je temperatura u peci? "))
print("Povecavam temperaturu!")
```

```
Potrebno je voditi racuna o temperaturi u peci.
Temperatura ne sme pasti ispod 50 celzijusovih stepeni.
Ako se to desi odmah se mora povecati temperatura u peci.
Kolika je temperatura u peci? 100
Temperatura je u pogodna za odrzavanje procesa.
Kolika je temperatura u peci? 120
Temperatura je u pogodna za odrzavanje procesa.
Kolika je temperatura u peci? 40
Povecavam temperaturu!
```

Задатак 042: Једноставна while петља

1. Помоћу while петље написати код који сабира вредност у промењивој број са 1 све док корисник не унесе као одговор на питање "не".
2. Додавати слово по слово у промењиву гес све док корисник не унесе слово "х".

Прожебати следеће задатке:

1. Сабирати све унете бројеве од стране корисника коришћењем петље while, све док је укупан збир тих бројева мањи од 10.
2. Корисник треба да уноси људска имена једно по једно све док не унесе име Jovan, када се while петља прекида.