

IZMENA VREDNOSTI PROMENLJIVIH

Измена вредности променљивих

Измена вредности скаларних променљивих

Начин на који смо до сада користили променљиве одговарао је ономе што си научио/научила из математике. Променљиве су имена које користимо да означимо одређене вредности и када су једном постављене оне се нису мењале током извршавања програма. Међутим, променљиве у програмима могу да мењају своју вредност! Посматрајмо следећи пример.

ПОКРЕНИ ПРОГРАМ

КОРАК ПО КОРАК

```
1 x = 3
2 print("x=", x)
3 x = 5
4 print("x=", x)
5
```

ActiveCode: 1 (promena_promenljivih_1)

Иста променљива `x` је на почетку програма имала вредност 3, то је исписано, а затим јој је промењена вредност на 5 и то је опет исписано. Изврши овај програм корак по корак и прати како се током извршавања мења вредност променљиве `x`.

Променљиве можемо схватити као чаше у којима су уписане одређене вредности. Да би се нов садржај могао сипати у чашу, стари мора да се проспе. Дакле, након доделе вредности променљивој (оператором `=`) стара вредност променљиве више није доступна. Ипак, аналогија са чашама није потпуно прецизна јер се након доделе облика `x = y` исти садржај налази у обе променљиве (чаша `y` из које се пресипа се не празни већ задржава свој садржај).

Размена вредности две променљиве

Размотримо како бисмо у програму разменили садржај две променљиве (на пример, x и y). Наиван начин би био да се то покуша на следећи начин:

ПОКРЕНИ ПРОГРАМ

КОРАК ПО КОРАК

```
1 # inicijalizujemo vrednost promenljivih
2 x = 3
3 y = 5
4
5 # ispisujemo vrednosti promenljivih
6 print("x=", x)
7 print("y=", y)
8
9 # razmenjujemo vrednost promenljivih - neispravno
10 x = y
11 y = x
12
13 # ispisujemo vrednosti promenljivih
14 print("x=", x)
15 print("y=", y)
16
```

ActiveCode: 1 (razmena_promenljivih_1)

Након доделе $x = y$ стара вредност променљиве x бива избрисана (оно што је у тој чаши било се просипа) и након те наредбе обе променљиве имају вредност 5 (вредност променљиве y).

ПОКРЕНИ ПРОГРАМ

КОРАК ПО КОРАК

```
1 # inicijalizujemo vrednost promenljivih
2 x = 3
3 y = 5
4
5 # ispisujemo vrednosti promenljivih
6 print("x=", x)
7 print("y=", y)
8
9 # razmenjujemo vrednost promenljivih
10 pomocna = x
11 x = y
12 y = pomocna
13
14 # ispisujemo vrednosti promenljivih
15 print("x=", x)
16 print("y=", y)
17
```

ActiveCode: 1 (razmena_promenljivih)

Размену вредности променљивих могуће је урадити и без помоћне променљиве, коришћењем распакивања парова тако што се изврши додела $(x, y) = (y, x)$. Ово решење је веома специфично за језик Python и не може се употребити у другим програмским језицима.

Размену вредности променљивих могуће је урадити и без помоћне променљиве, коришћењем распакивања парова тако што се изврши додела $(x, y) = (y, x)$. Ово решење је веома специфично за језик Python и не може се употребити у другим програмским језицима.

ПОКРЕНИ ПРОГРАМ

КОРАК ПО КОРАК

```
1 # inicijalizujemo vrednost promenljivih
2 x = 3
3 y = 5
4
5 # ispisujemo vrednosti promenljivih
6 print("x=", x)
7 print("y=", y)
8
9 # razmenjujemo vrednost promenljivih
10 (x, y) = (y, x)
11
12 # ispisujemo vrednosti promenljivih
13 print("x=", x)
14 print("y=", y)
15
```

Промена стања на банковном рачуну

Нова вредност променљиве може да зависи од њене старе вредности. На пример, променљива у програму може представљати стање (износ новца) на нечијем банковном рачуну. Када се изврши уплата ново стање се израчунава као збир претходног стања и износа који је уплаћен.

ПОКРЕНИ ПРОГРАМ

КОРАК ПО КОРАК

```
1 stanje = 17352
2 print("stanje=", stanje)
3
4 # Uplaćujemo na račun 3400 dinara
5 stanje = stanje + 3400
6 print("stanje=", stanje)
7
8 # Uplaćujemo na račun 2200 dinara
9 stanje = stanje + 2200
10 print("stanje=", stanje)
11
12 # Podižemo sa računa 1500 dinara
13 stanje = stanje - 1500
14 print("stanje=", stanje)
15
```

Приметимо да у математици једнакост $stanje = stanje + 3400$, ако се схвати као једначина, нема решења. Међутим, ово није математичка једначина, већ је наредба доделе, специфична за програмирање. Овом наредбом доделе се постиже да стање на рачуну после ње буде једанко стању на рачуну пре ње увећаном за 3400 динара (тј. да је вредност променљиве након извршавања ове наредбе доделе једнака збиру вредности променљиве пре доделе и броја 3400). Математички гледано, једнакост која описује ову доделу је заправо $stanje_{posle} = stanje_{pre} + 3400$, што има много више смисла.

Постоје облици програмирања (на пример, такозвано чисто функционално програмирање) и програмски језици у којима измена вредности променљивих на овај начин није допуштена.

Измена вредности листе

За разлику од торки и ниски које су непроменљиве структуре података, листе и речници су променљиви.

Посматрајмо наредни програм.

ПОКРЕНИ ПРОГРАМ

КОРАК ПО КОРАК

```
1 heroj = "Supermen"  
2 print(heroj)  
3 heroj = "Betmen"  
4 print(heroj)  
5
```

Supermen

Betmen

Иако изгледа да је ниска `heroj` промењена, заправо је током извршавања програма та променљива садржала две потпуно различите ниске ("`Supermen`", па онда "`Betmen`"). Када кажемо да су ниске непроменљиве, то значи да не можемо да им мењамо, додајемо или бришемо слова (на пример, од ниске "`Supermen`" не можемо направити ниску "`Superman`" тако што бисмо изменили само слово `e` у слово `a`).

Измена кошаркаша



Мирослав Радуљица се повредио и уместо њега наступа Огњен Кузмић. Представи ту измену у програму.

ПОКРЕНИ ПРОГРАМ

КОРАК ПО КОРАК

```
1 igraci = ["", "Стефан Јовић", "Марко Симоновић", "Богдан Богдановић", \  
2          "Никола Калинић", "Милан Мачван", "Стефан Марковић", \  
3          "Немања Недовић", "Мирослав Радуљица", "Милош Теодосић", \  
4          "Никола Јокић", "Владимир Штимац", "Стефан Бирчевић"]  
5 print(igraci[8])          # на poziciji 8 је Raduljica  
6 igraci[8] = "Огњен Кузмић" # на poziciju 8 postavljamo Kuzmića  
7 print(igraci[8])          # sada је на poziciji 8 Kuzmić  
8
```

Учитавање листе



Напиши програм који учитава листу 5 бројева и на крају исписује њен садржај.

ПОКРЕНИ ПРОГРАМ

КОРАК ПО КОРАК

```
1 brojevi = []
2 for i in range(5):
3     brojevi.append(int(input()))
4 print(brojevi)
5
```

ActiveCode: 1 (учитавање листе)

Други начин би био да направимо листу која садржи 5 нула, а затим да јој током учитавања мењамо један по један елемент.

ПОКРЕНИ ПРОГРАМ

КОРАК ПО КОРАК

```
1 brojevi = [0] * 5
2 for i in range(5):
3     brojevi[i] = int(input())
4 print(brojevi)
5
```

Трећи начин, можда и најелегантнији, заснива се на компрехенсији, о којој ће бити речи касније.

ПОКРЕНИ ПРОГРАМ

КОРАК ПО КОРАК

```
1 brojevi = [int(input()) for i in range(5)]  
2 print(brojevi)  
3
```

ActiveCode: 1 (учитавање листе 2)

Промена цене аутомобила

ПОКРЕНИ ПРОГРАМ

КОРАК ПО КОРАК

```
1 cene_automobila = {"fiat 500l": 11990,  
2                   "renault clio": 9650,  
3                   "toyota corolla": 13990}  
4 print(cene_automobila["fiat 500l"])  
5 cene_automobila["fiat 500l"] = 11500  
6 print(cene_automobila["fiat 500l"])  
7
```

ActiveCode: 1 (промена цене аутомобила)

Додавање новог елемента у речник вршимо на веома сличан начин као и промену постојеће вредности.

ПОКРЕНИ ПРОГРАМ

КОРАК ПО КОРАК

```
1 cene_automobila = {"fiat 500l": 11990,  
2                   "renault clio": 9650,  
3                   "toyota corolla": 13990}  
4 cene_automobila["opel astra"] = 12677  
5 print("toyota corolla: ", cene_automobila["toyota corolla"])  
6 print("opel astra: ", cene_automobila["opel astra"])  
7
```