



Курс програмског језика Пајтон ниво 0

03 час – промењиве

1. Садржај

Садржај..... 1

Кратак опис 1

Идентификатор и оператор доделе..... 1

Бројеви 3

Математички оператори 4

Тест..... 5

Вежбања 6

Задаци 7

2. Кратак опис

Програми обично смештају податке у меморију рачунара и изводе операције над тим подацима. Програми користе **промењиве** да би приступали и манипулисали са подацима који су у рачунарској меморији. Када промењива прикаже вредност из меморије рачунара каже се да промењива означава (*references*) ту вредност.

3. Идентификатор и оператор доделе



Промењива **a** указује на меморијску локацију у којој се налази вредност **23**. То је постигнуто помоћу **оператора доделе** (=, знак једнакости) (*assignment operator*). Промењиве се представљају помоћу **идентификатора**.



Промењива **a** указује на меморијску локацију у којој се налази вредност **23**, промењива **b** указује на локацију са вредности **11**, промењива **name** указује на локацију са вредности **GOOD**.

a = 23 b = 11 name = "GOOD"

**Пример 01) Додела вредности у интерпретеру**

```
>>> a = 23
>>> b = 11
>>> name = "GOOD"
>>> print(a, b, name)
23 11 GOOD
```

У сваком новом реду се укуца нови идентификатор. Промењивој са укуцаним именом идентификатора се помоћу оператора доделе, додели вредност. Функцијом *print* се испише у новом реду садржај меморијских локација на које упућују промењиве.



Ако промењиве већ упућују на одређене вредности могуће је да се промене меморијске локације на које ће те исте промењиве да упућују. На тај начин промењиве упућују на неке друге вредности.

На слици, промењива **b** сада упућује на другу меморијску локацију, у којој се налази вредност 8.

Пример 02) Промена вредности промењиве

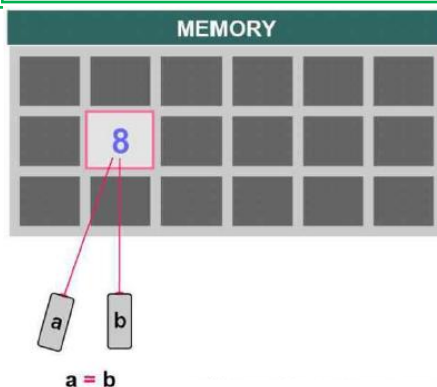
```
>>> a = 23
>>> b = 11
>>> name = "GOOD"
>>> print(a, b, name)
23 11 GOOD
>>> b = 8
>>> print(a, b, name)
23 8 GOOD
```

Промењиве **a**, **b** и **name** указују на меморијске локације као у претходном примеру.

Ако се помоћу оператора доделе, промењивој **b** "додели" нова вредност, промењива указује на нову локацију са вредности 8.

НАПОМЕНА

Промењива садржи њој прво додељену вредност све док се промењивој не додели нека друга вредност. Прво додељена вредност више није у употреби па Пајтон интерпретер аутоматски је брише из меморије преко процеса који се зове **сакупљање ђубрета** (*garbage collection*), чиме се ослобађа меморијска локација за даље коришћење.



Када се промењивој додељује вредност коју већ има нека друга промењива, интерпретер ће упутити да обе промењиве показују на исту меморијску локацију са жељеном вредности.

На овај начин се штеди употреба меморијских локација.

**Пример 03) Додела истих вредности**

```
>>> a = 11
>>> b = 8
>>> print(a, b)
11 8
>>> a = 8
>>> print(a, b)
8 8
```

Променљиве **a** и **b** имају различите вредности.

Коришћењем оператора доделе, променљивој **a** је променљива вредност.

Сада, променљиве **a** и **b** указују на исту меморијску локацију, тј. имају исте вредности.

Пример 04) Провера указивања на исту локацију

Постоје два начина за проверу да ли променљиве указују на исту меморијску локацију:

```
>>> a = 8
>>> b = 8
>>> print(id(a), id(b))
1650341760 1650341760
```

Први је коришћењем инструкције **id(a)** унутар функције **print**, чиме се на екрану исписује адреса меморијске локације на коју упућује променљива **a**.

```
>>> a = 8
>>> b = 8
>>> a is b
True
```

Други је коришћењем логичког оператора **is**.

Оператор упоређује адресе меморијских локација на које упућују променљиве **a** и **b** и ако су адресе идентичне, даје као резултат тачно (**True**) а у супротном нетачно (**False**).

НАПОМЕНА**Правила за давање имена променљивима (избор идентификатора):**

- не може се користити кључна реч
- у имену не може постојати празно место
- први карактер може бити било које велико или мало слово или доња линија (**_**) (*underscore*)
- следећи карактери у имену могу бити слова, цифре или доње линије
- постоји разлика између малих и великих слова (*ja* није иста променљива као *Ja* или *jA*)
- име променљиве би требало да има описно значење о сврси променљиве (*tezina_u_uncama*, *baza_podataka_ucenici*, *tabela_repertoar_bioskopa*)

4. Бројеви**НАПОМЕНА**

У Пајтону постоји више типова променљивих. Типови променљивих се деле на нумеричке (бројчане), стрингове, листе (*lists*), туплете (*tuples*) и речнике (*dictionary*). Најчешће се користе стрингови и два типа бројчаних променљивих : реални и цели бројеви.

За смештање реалних бројева (бројеви са децималним зарезом) у меморију, рачунар користи другачије технике него за смештање целих бројева. Из тог разлога Пајтон користи типове података за разликовање типова вредности у меморији.

Када се цео број смести у меморији, Пајтон га описује као **int**, док реалан број описује као **float**.



НАПОМЕНА

Број унет у код програма се назива бројчани литерал (*numeric literal*).

Када Пајтон интерпретер прочита бројчани литерал у програмском коду, он одређује тип података према правилима:

- бројчани литерал написан као цео број је типа *int* (7, 124, -8)
- бројчани литерал написан са децималном тачком је типа *float* (1.5, -5.0)

Пајтон интерпретер мора одмах да дефинише бројчани тип података за вредност у питању да би могао одредити са каквим операцијама може да ради.

Пример 05) Дефинисање бројчаног типа података

```
print("Koji je tip podataka broj 1?")
print(type(1))
print("Koji je tip podataka broj 1.00?")
print(type(1.00))
input("ENTER...")
```

```
Koji je tip podataka broj 1?
<class 'int'>
Koji je tip podataka broj 1.00?
<class 'float'>
ENTER...
>>> |
```

У примеру је приказано како је Пајтон дефинисао унету бројчану вредност помоћу уграђене (built-in) функције **type** у скрипт моду. Порука се исписује у следећој линији, **<class 'int'>**, и каже да је вредност 1 типа **int**, целобројни број. За вредност 1.00 порука каже да је то реални број, типа **float**.

5. Математички оператори

симбол	операција	опис
+	сабирање	сабира два броја
-	одузимање	одузима два броја
*	множење	множи два броја
/	дељење	дели два броја и као резултат даје реалан број
//	целобројно дељење	дели два броја и као резултат даје цели број
%	остатак дељења	приказује остатак дељења два броја
**	степеновање	диже број на дати степен

Пример 06) Математички изрази

```
a = 3
b = a + 1
print("Posle operacije sabiranja b=",b)
b = a - 1
print("Posle operacije oduzimanja b=",b)
b = a * 2
print("Posle operacije mnozenja b=",b)
b = a / 2
print("Posle operacije deljenja b=",b)
b = a // 2
print("Posle operacije celobrojnog deljenja b=",b)
b = a % 2
print("Posle operacije moduo - ostatak deljenja b=",b)
b = a ** 2
print("Posle operacije stepenovanja b=",b)|
```



Posle operacije sabiranja b= 4
 Posle operacije oduzimanja b= 2
 Posle operacije mnozenja b= 6
 Posle operacije deljenja b= 1.5
 Posle operacije celobrojnog deljenja b= 1
 Posle operacije modulo - ostatak deljenja b= 1
 Posle operacije stepenovanja b= 9

Приоритети означавају да приликом израчунавања математичких израза, Пајтон користи правила приоритета, што значи да неке математичке операције имају предност над другим математичким операцијама. Из тог разлога ће се неке математичке операције (оне са већим приоритетом) увек извршавати пре других (са мањим приоритетом).

Од до сада коришћених математичких операција, по приоритету оне су поређане:

виши приоритет	симбол	операција
1	()	заграде
2	**	степеновање
3	*	множење, дељење, целобројно дељење
	/	
	//	
4	+	сабирање, одузимање
	-	
нижи приоритет	симбол	операција

Математичке операције истог приоритета се извршавају редом са лева на десно.

Пример 07) Приоритети

```

>>> 2 + 3 * 2
8
>>> 2 + 2 + 2 * 0
4
>>> 2 - 2 * 2
-2
>>> (2 + 2) * (4 * 4 - (2 * 6 + 4))
0
>>> 10 - 3 ** 2
1
>>> 21 % 4 + 2 ** 2
5
  
```

- 1) Множење пре сабирања
- 2) Множење пре сабирања
- 3) Множење пре одузимања
- 4) Прво се реализује најдубља заграда, затим се нађе резултат у првој па у другој загради, па се изврши множење
- 5) Степеновање пре одузимања
- 6) Модуо пре степеновања затим сабирање

Тест

1. Шта користе програми за рад са подацима ?
2. Где су смештене промењиве ?
3. Објаснити шта промењива означава.
4. Шта је то оператор доделе ?
5. Шта је то идентификатор ?
6. Која функција се користи за приказ садржаја промењиве ?
7. Промена вредности у промењивој значи указивање на другу меморијску локацију?
8. Објаснити процес "сакупљања ђубрета".



9. Да ли два различита идентификатора могу указивати на исту меморијску локацију?
10. Која инструкција се користи за проверу адресе на коју упућује промењива ?
11. Чему служи логички оператор *is* ?
12. Који су могући резултати рада логичког оператора *is* ?
13. Навести сва правила избора идентификатора.
14. Навести типове промењивих у Пајтону.
15. Како се другачије називају бројеви са децималним зарезом у Пајтону ?
16. Да ли је неопходно одмах описати у Пајтону врсту броја пре његове употребе ?
17. Шта се користи за опис типа броја у Пајтону ?
18. Како се називају бројеви у коду програма ?
19. Да ли је Пајтон интерпретер способан да сам открије који је тип унетог броја ?
20. Зашто је неопходно дефинисати тип бројчаног податка ?
21. `price = 99.0`. Који је тип податка вредност на коју упућује промењива `price` ?
 - а) `int`
 - б) `float`
 - в) `currency`
 - г) `str`
22. Шта су то `built-in` функције ?
23. Која се `built-in` функција користи за испис типа бројчаног податка ?
24. Навести све приказане математичке операције у Пајтону.
25. Навести симболе за приказане математичке операције у Пајтону.
26. У чему је разлика између реализације `/` и `//` операције ?
27. Којом операцијом се добија кореновање у Пајтону ?
28. Шта је то правило приоритета у Пајтону ?
29. Које три операције, од до сада приказаних, имају исти приоритет ?
30. Поређати приказане математичке операције по приоритету.
31. Како се извршавају операције истог приоритета ?

Вежбања

1. Употребом интерпретера доделити промењивој *novo* бројчану вредност 4 а промењивој *stara* бројчану вредност 10.
2. Проверити коришћењем функције `id()` да ли промењиве *novo* и *stara* указују на исту меморијску локацију.
3. Доделити промењивој *nase* вредност "zlato" а промењивој *vase* вредност "srebro".
4. Проверити коришћењем логичког оператора `is` да ли промењиве *nase* и *vase* указују на исту адресу меморијских локација.
5. Доделити промењивој *malo* бројчану вредност 4 а промењивој *veliko* бројчану вредност 4.00.
6. Проверити коришћењем функције `id()` да ли промењиве *malo* и *veliko* указују на исту меморијску локацију.
7. Проверити коришћењем логичког оператора `is` да ли промењиве *malo* и *veliko* указују на исту адресу меморијских локација.
8. Проверити да ли промењиве *novo* и *malo* указују на исту меморијску локацију.



9. Доделити промењивој *novo* вредност 5. Да ли је извршен процес “сакупљања ђубрета” за меморијску локацију са вредности 4 ? Ако не зашто не ?
10. Доделити промењивој *malo* вредност 6. Да ли је извршен процес “сакупљања ђубрета” за меморијску локацију са вредности 4 ? Ако не зашто не ?
11. Проверити којих су типова до сада коришћене промењиве (*novo*, *staro*, *nase*, *vase*, *malo*, *veliko*).
12. Под претпоставком да су промењиве **rezultat**, **w**, **x**, **y**, **z** сви типа целих бројева, и да $w = 5$, $x = 4$, $y = 8$, $z = 2$. Које вредности ће бити смештене у промењиву **rezultat** после извршавања сваке од следећих израза ?
`rezultat = x + y`
`rezultat = z * 2`
`rezultat = y / x`
`rezultat = y - z`
`rezultat = w // z`
13. Напиши израз који додељује збир 10 и 14 промењивој *suma*.
14. Напиши израз који одузима промењиву *placeno* од промењиве *zarada* и резултат додељује промењивој *ostatak*.
15. Напиши израз који множи промењиву *podrezultat* са 0.15 и додељује резултат множења промењивој *rezultat*.
16. Шта ће се појавити као резултат на екрану ?
`a = 5`
`b = 2`
`c = 3`
`rezultat = a + b * c`
`print(rezultat)`
17. Шта ће се појавити као резултат на екрану ако се у претходном примеру измени само: `rezultat = a + (b * c)` ? Зашто су оба примера са истим резултатом ? На који начин променити исту линију да би се прво извело сабирање ?

Задаци

1. Укупна куповина

Купац у продавници купује пет артикала. Напиши програм који приказује цене за сваки артикал а затим приказује суму цена, порез на куповину и укупну уплаћену суму. Претпоставка је да је порез на куповину 7 процената.

2. Пређена раздаљина

Растојање које ауто прелази је: $\text{rastojanje} = \text{brzina} * \text{vreme}$

Ауто путује 70 километара на сат. Напиши програм који приказује:

- растојање које ће ауто прећи за 6 сати
- растојање које ће ауто прећи за 10 сати
- растојање које ће ауто прећи за 15 сати

3. Километри по литри

Километара по литри (KPL) за један ауто се може израчунати као:

$\text{KPL} = \text{broj kilometara} / \text{kolicina korišćenog goriva u litrima}$

Напиши програм који рачуна колико километара прелази ауто при потрошњи једног литра горива. Код треба да израчуна KPL и прикаже резултат.



4. Напојнице, порези и сума

Напиши програм који рачуна укупан број оброка наручених у ресторану. Програм има фиксну цену оброка, број продатих оброка, напојнице за келнере од 18 процената и 7 процената за порез. Приказати сваки од ових резултата и на крају укупни остатак новца који остаје ресторану.

5. Конвертор температуре

Напиши програм који претвара целзијусове степене у фанерхајтове степене према формули: $F=(9*C/5)+32$

Као улаз у коду је температура у цезијусовим степенима и на излазу приказује температуру у фанерхајтовим степенима.

6. Процена дечака и девојчица

Напиши програм који израчунава број дечака и девојчица у одељењу у процентима.

$$F = \frac{9}{5}C + 32$$