



## Курс програмског језика Пајтон ниво 0

## 04 час – конверзија

## 1. Садржај

Садржај.....	1
Кратак опис .....	1
Конверзија.....	1
Додатни оператори .....	3
Случајни бројеви .....	4
Тест.....	4
Вежбања .....	5
Задаци .....	7

## 2. Кратак опис

Због потреба кода, понекад је неопходно извршити промену типа података. Промена типа података се назива **конверзија**. Пајтон омогућава да конверзија буде трајна или привремена.

## 3. Конверзија

Трајна конверзија се може извршити додељивањем промењивој другог типа вредности од онога који је оригинално био додељен.

```
>>> a = 1
>>> print(type(a))
<class 'int'>
>>> a = 1.00
>>> print(type(a))
<class 'float'>
>>> a = 'pajton'
>>> print(type(a))
<class 'str'>
```

Доделом различитих типова вредности промењива мења своју дефиницију, декларише се према потребама кода програма. У Пајтону није неопходно декларисати промењиву, посебно ако је сасвим јасно по типу додељене вредности којег ће типа бити промењива.

Привремена конверзија омогућава промену типа промењиве без експлицитне доделе вредности промењивој.

```
>>> a = 1
>>> print(a)
1
>>> print(float(a))
1.0
>>> print(a)
1
```

Види се да је оригинално додељена вредност промењивој **a** била типа целог броја (`int`). Привременом конверзијом је натерана промењива **a** да промени тип података у реални тип. Истовремено је и њен податак постао типа реалног броја. Ова конверзија је важећа само у том тренутку, у тој линији кода, јер већ у следећој линији се види да промењива **a** је задржала свој првобитни тип података.

\*Проблем може представљати унос бројчаних вредности са тастатуре помоћу функције `input` јер ова функција увек враћа кориснички улаз као стринг.

Уграђене функције које конвертују стринг у бројчане типове су:

`int(argument)` - спровођењем аргумента у функцију `int()`, она враћа конвертовану вредност аргумента у `int`;

`float(argument)` - спровођењем аргумента у функцију `float()`, она враћа конвертовану вредност аргумента у `float`;



```
string_vrednost = input("Koliko sati si radio? ")  
sati = int(string_vrednost)
```

Првом линијом се додељује унета стринг вредност у промењиву *string\_vrednost*. Друга линија позива *int()* функцију и спроводи *string\_vrednost* као аргумент. Вредност означена са *string\_vrednost* је конвертована у *int()* и додељена промењивој *sati*.

Други начин конвертовања стринга у целобројну вредност:

```
sati = int(input("Koliko sati si radio? "))
```

Овај исказ користи угњеждене позиве функције (*nested function calls*). Вредност која је враћена са улаза се прослеђује као аргумент у функцију *int()*.

```
# unos.py  
# prikazuje unete vrednosti  
# Rankovic Nebojsa 25.12.2015.
```

```
# uneti korisnikove podatke  
ime = input('Kako se zoves? ')  
godine = int(input("Koliko imas godina? "))  
plata = float(input("Kolika ti je plata?"))  
  
# prikaz podataka  
print("Uneti su sledeci podaci:")  
print("Ime:", ime)  
print("Godine:", godine)  
print("Plata:", plata)  
input("ENTER...")
```

```
Kako se zoves? Bil  
Koliko imas godina? 50  
Kolika ti je plata?100000  
Uneti su sledeci podaci:  
Ime: Bil  
Godine: 50  
Plata: 100000.0  
ENTER...  
>>> |
```

Конвертовање из стринга у бројчане вредности има смисла само ако су у стринг уписане само цифре. Ако то није случај, јавља се грешка позната као изузетак (*exception*). То је неочекивана грешка која се јавља док програм ради, чиме се програм зауставља.

\* Када се изводи математичка операција на два **операнда** (вредности или израза), тип податка резултата ће зависити од типа података операнада. Правила рачунања математичких израза:

- када се операција изводи са две цеобројне вредности резултат ће бити целобројан
- када се операција изводи са две реалне вредности, резултат ће бити реална вредност
- ако се операција изводи са једном целобројном и једном реалном вредности, целобројна вредност ће бити привремено конвертована у реалну вредност и резултат ће бити реална вредност

Конверзија у последњем правилу је имплицитна. Ако је потребна експлицитна конверзија, користи се *int()* или *float()* функција:

```
float_vrednost = 2.6  
inv_vrednost = int(float_vrednost)  
print(inv_vrednost)  
input("ENTER...")
```

```
2  
ENTER...  
>>>
```

У примеру, у другој линији кода се врши придруживање *float\_vrednosti* као аргумента *int()* функције. Ова функција враћа вредност 2, која се додељује промењивој *int\_vrednosti*.



#### 4. Додатни математички оператори

\*Скраћени облик писања основних математичких оператора:

нормалан облик	скраћени облик
$a = a + 1$	$a += 1$
$a = a - 5$	$a -= 5$
$a = a * 100$	$a *= 100$
$a = a / 2$	$a /= 2$

\*Релациони оператори

оператор	значење
>	веће од
<	мање од
>=	веће или једнако
<=	мање или једнако
==	једнако (идентично са)
!=	није идентично са

Ово су оператори који испитују тачност или нетачност претпостављених релација између вредности или израза. Резултат ових оператора је тачно (True) или нетачно (False).

```
>>> a = 10
>>> b = 11
>>> print(a > b)
False
>>> print(a < b, b >= a)
True True
>>> c = 10
>>> print(a == c)
True
>>> print(b != c)
True
```

Оператор испитивања идентичности (==) није исто што и оператор доделе (=). Оператор испитивања идентичности је релацијски оператор и са њиме се упоређују вредности или изрази. Ако су вредности или изрази једнаки, резултат упоређивања је тачан (True).

Супротно њему, оператор неједнакости (!=) ће дати да је резултат тачан (True) ако вредности или изрази нису идентични.

\*Логички оператори

Ово су оператори који повезују релационе операторе или изразе у један сложенији логички израз. Резултат ових оператора је тачно (True) или нетачно (False).

**and** – “и” оператор спаја два релациона оператора или израза у један сложенији израз. Оба релациона оператора или израза морају бити тачни да би цео сложени израз био тачан.

**or** – “или” оператор спаја два релациона оператора или израза у један сложенији израз. Довољно је да је само један од два релациона оператора или израза тачан да би цео сложени израз био тачан.

**not** – “не” оператор је **унарни** оператор који ради са једним операндом (релационим оператором или изразом). Претвара резултат операнда у његову супротност, ако је операнд тачан даје као резултат нетачно.

```
>>> a = 1
>>> b = 0
>>> print(a > b)
True
>>> print((a > b) and (a <= b))
False
>>> print((a == b) or (a >= b))
True
>>> print(not(a > b))
False
```

У датом примеру се користе логички оператори над релационим операторима.

Резултат је нетачан јер други оператор даје нетачан резултат.

Овде су оба оператора тачна и резултат је исти.

Овде је резултат супротан од резултата који даје сам оператор.



## 5. Случајни бројеви

Генерисање случајног броја у задатом опсегу вредности је у сваком програмском језику задатак посебне функције. Таква функција даје резултат коришћењем унапред дефинисаног алгоритма па се не може рећи да је тако добијени број потпуно случајан (тако добијене вредности су псеудослучајне).

Функција се назива **random**.

```
import random
d = random.randrange(6) + 1
print(d)
input("ENTER za kraj...")
```

6  
ENTER za kraj...  
1  
ENTER za kraj...  
2  
ENTER za kraj...

Инструкција **import** омогућава учитавање и коришћење модула **random** у датом коду.

### НАПОМЕНА

Модули су фајлови са кодом који се могу користити у било којим другим програмима. Најчешће у себи садрже колекцију кодова за употребу у специфичним областима. Тако **random** модул садржи функције повезане са генерисањем случајних (*random*) бројева и резултата.

Модул **random** омогућава коришћење функције **randrange()**. Ова функција се позива као: `random.randrange(6)` и тако генерише случајан број. У загради се ставља цео број који представља горњу границу опсега генерисања броја. У овом случају се генерише случајан број у опсегу од 0 до 5 (не и 6). Сабирањем добијеног резултата са 1 се омогућава да опсег добијеног броја буде од 1 до 6.

### Тест

1. Шта означава реч конверзија ?
2. Зашто се конверзија примењује у програмирању ?
3. Шта је то трајна конверзија ?
4. Како се врши трајна конверзија ?
5. Да ли се трајном конверзијом мења тип већ постојећег податка ?
6. Шта се у ствари мења са привременом конверзијом ?
7. У који тип података се конвертује колриснички улаз коришћењем функције **input**?
8. Како се врши конверзија садржаја промењиве у целобројни тип ?
9. Како се врши конверзија садржаја промењиве у реални тип ?
10. Како се врши конверзија садржаја промењиве у стринг тип ?
11. Шта се дешава коришћењем следеће линије кода:  
`a = int(input("Unos: "))`
12. Шта се сматра под изразом nested function calls ?
13. Када се јавља грешка под именом exception ?
14. Шта је то операнд ?
15. Која су правила рачунања математичких израза ?
16. У ком случају рачунања математичког израза долази до привремене конверзије?
17. Која конверзија уноси грешку у рачунању ?



18. Који математички оператори имају скраћени облик представљања ?
19. Да ли се следећи израз може написати у скраћеном облику и зашто:  $a=b+2$  ?
20. Чему служе релацијски оператори ?
21. Који су релациони оператори до сада приказани ?
22. Какви су резултати рада ових оператора ?
23. У чему је разлика између оператора  $==$  и  $=$  ?
24. Како се користе логички оператори ?
25. Навести све логичке операторе.
26. Шта значи унарни оператор ?
27. Који од логичких оператора је унарни ?
28. Да ли у Пајтону постоји наредба која генерише случајан број ?
29. За добијање случајно генерисаног броја се користи :
  - а) логички оператор
  - б) алгоритам
  - в) стринг промењива
  - г) функција `print`
30. Са којом инструкцијом се учитавају модулу у Пајтону:
  - а) `import`
  - б) `random`
  - в) `randrange`
  - г) `input`
31. Шта су то модули ?
32. Како се назива модул који се користи при генерисању случајних бројева ?
33. Како се назива функција помоћу које се генерише случајан број ?
34. Шта је резултат следеће линије кода:  
`random.randrange(2)`
35. Које све вредности је могуће генерисати са наредбом из претходног задатка ?

### Вежбања

1. Промењиве **a**, **b** и **c** имају вредности 1, 2 и 3 редом. Проверити типове промењивих.
2. Променити вредности претходних промењивих у 1.00, 2.00 и 3.00. Проверити типове промењивих.
3. Коришћењем привремене конверзије променити тип промењиве **a** у целобројну а затим и у стринг. Проверити типове промењивих.
4. Проверити који је тип промењиве којој се додели унос од стране корисника.
5. Променити тип исте промењиве у целобројну.
6. Променити тип исте промењиве у реалну.
7. Доказати појаву грешке изузетак.
8. Који ће бити тип резултата следећих израза:  
 $a = 2, b = 2.00$ 
  - а)  $c = a + a$
  - б)  $c = a**2$
  - в)  $c = a - b$
  - д)  $a += 1$
  - е)  $c = a//1.0$
  - ф)  $c = a//1$
  - г)  $b = b - a$
  - х)  $a /= b$
9. Коришћењем скраћеног облика писања математичких оператора написати следеће изразе:



- a)  $a = 5$   
 $a = a + 1$
- b)  $a = 10$   
 $b = a + 1$
- c)  $a = 2.00$   
 $b = 4.00$   
 $b = a - b$

10. Испитати следеће релације:

$a = 1, b = 2, c = 2.5$

- a) **a** веће од **b**
- b) **b** мање или једнако **c**
- c) **c** идентично са 2.5
- d) **c** веће или једнако од **a** плус **b**
- e) **b** идентично са конвертовано **c** у целобројно

11. Шта је резултат:  $0 \neq \text{int}(0.1)$

12. Који логички оператор одговара којој табели:

a	b	?
0	0	0
0	1	0
1	0	0
1	1	1

a	b	?
0	0	0
0	1	1
1	0	1
1	1	1

a	?
0	1
1	0

13. Какви су резултати следећих логичких израза:

$a = 1, b = 100$

- a)  $(a \geq 100) \text{ or } (a \geq b)$
- b)  $(a \geq 100) \text{ and } (a \geq b)$
- c)  $\text{not}((a \geq 100) \text{ and } (a \geq b))$

14. У следећим примерима ако је потребно, променити само један логички оператор у други логички оператор да би се добио као резултат True:

- a)  $(1 == 1) \text{ and } (0 != 0)$
- b)  $\text{not}((5 \geq 5.1) \text{ and } (\text{float}(2) == \text{int}(1.99)))$
- c)  $(1 != 0) \text{ not } (2 == \text{int}(2.99))$

15. Преправити следећу линију кода тако да се генеришу случајни бројеви у опсегу од 0 до 10:  $a = \text{random.randrange}(9)$



16. Преправити следећу линију кода тако да се генеришу случајни бројеви у опсегу од -1 до 8: `a = random.randrange(5)`

### Задаци

**1. Улаз на куб**

Написати код програма којим се вредност унешена од стране корисника диже на куб и резултат приказује на екрану.

**2. Улаз и математика**

За две унешене реалне бројчане вредности од стране корисника израчунати збир, разлику, производ и количник.

**3. Логичко и**

Нацртати табелу за логички оператор *and* и при томе користити логички оператор *and*.

**4. Корисников случајан избор**

Нека корисник сам изабере горњу границу опсега за генерисање случајног броја и приказати на екрану добијени резултат генерисања.

**5. Бацање коцкица**

Написати код програма који симулира бацање две класичне коцкице, сабира добијене резултате бацања и приказује збир.