

Python – основне аритметичке операције

Рачунар или компјутер (енгл. computer) је справа која рачуна, тј. справа која је направљена тако да може веома брзо и ефикасно да изводи рачунске операције над бројевима. Упознавање са програмским језиком Python ћемо почети управо од аритметичких операција и изачунавања вредности аритметичких израза.

Да би могао да испробаш примере у наставку текста и провериш сопствена решења биће ти потребно неко од програмских окружења Python-а. У којим све окружењима можеш покретати Python програм и како их можеш користити на свом рачунару сазнаћеш када одгледаш следећи видео.

Python - инсталација

Велика помоћ у учењу биће ти интерактивни уџбеник који се налази на адреси

http://petlja.org/BubbleBee/r/kursevi/interactive-book-py6

Уџбеник нуди различите могућности за проверу знања, садржи велики број питања и задатака. Програмске кодове који представљају решења задатака је могуће покренути и тестирати и у окружењу самог уџбеника.

Сабирање, одузимање, множење

У овој лекцији ћемо се бавити изразима који садрже операције сабирања, одузимања и множења, које се у програмском језику Python употребљавају на следећи начин:

- Збир бројева, на пример 3 и 5, бележи на готово идентичан начин као у математици, `3 + 5`.
- Разлика бројева, на пример 8 и 2, се, такође бележи идентично математичком запису, `8 - 2`.
- Производ бројева, у математичком запису $4 \cdot 6$, се у Python-у записује са `4 * 6`.

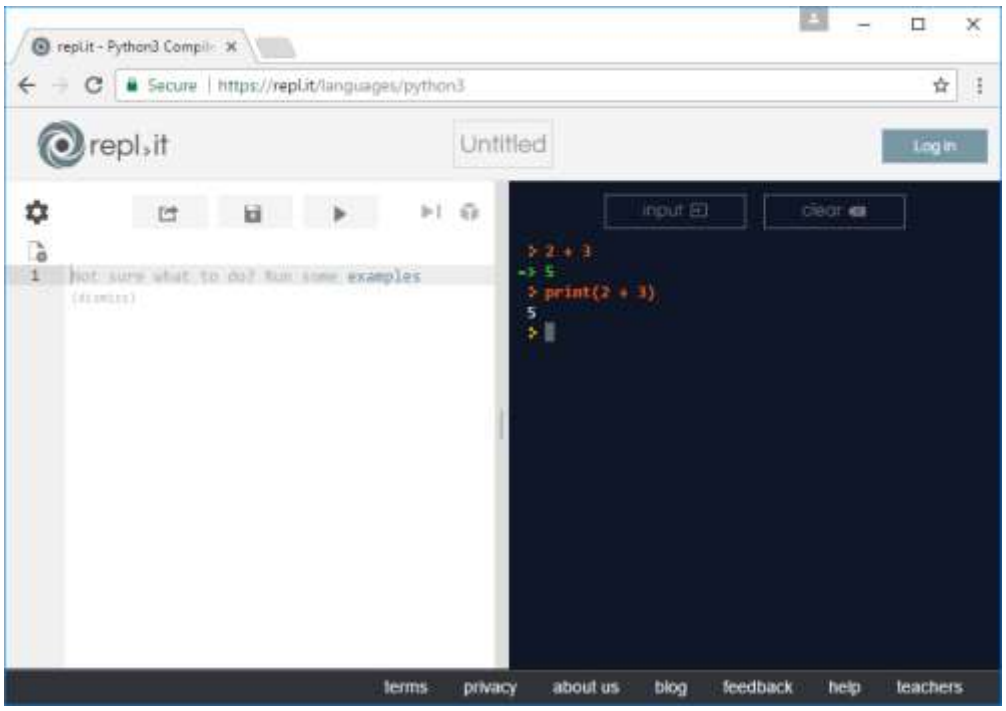


Програмски језик Python, наравно, уме и да дели, да израчунава остатак при дељењу и цео део количника и много штошта друго. О овим операцијама ћемо говорити на неком од наредних часова.

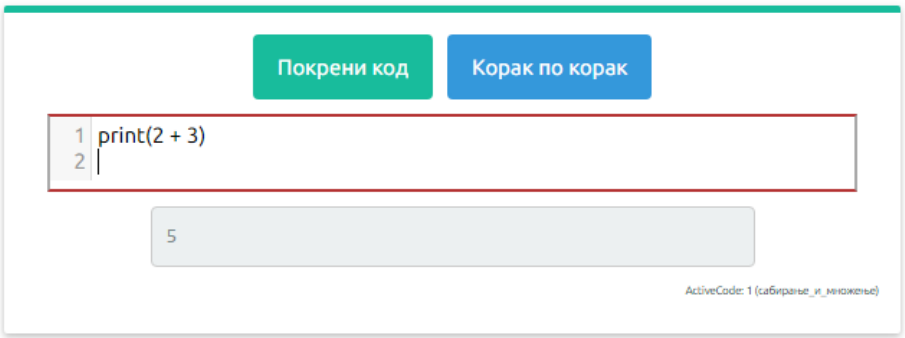
Како се у Python програмском окружењу добија вредност аритметичког израза? Погледај видео лекцију

Python - аритметичке операције

Покрени програмско окружење Python-а. Провери коју ће вредност израза, нпр. 2 + 3, окружење дати за тако што ћеш након откуцаног изараза притиснути тастер Enter.



Исти резулат ћеш добити и када откуцаш `print(2+3)`. Такав запис ће ти бити потребан уколико желиш да га извршиш у окружењу интерактивног уџбеника.



Задатак 1.
У програмском окружењу Python-а израчунај вредности израза:

1. 3 + 5
2. 4 * 6

Приметимо да смо око оператора куцали размаке (на пример, 3 + 5, 8 - 2 или 4 * 6). Ти размаци нису неопходни и исправно је написати и 3+5 или 4*6. Програмери воле да куцају размак око сваког оператора (знака операције) да би добили текст програма који лепше изгледа и који се лакше може прочитати.

Сложени изрази, приоритет оператора и заграде

Некада је потребно да решавамо задатке који укључују више рачунских операција и тада можемо користити сложеније изразе, потпуно исто како смо навикли у математици. На пример, посматрајмо следећи задатак који је преузет из једне збирке задатака из математике:

Израчунај производ збира бројева 874 и 437 и разлике бројева 915 и 364.

Уз помоћ рачунара можеш веома једноставно решити овај задатак (чак једноставније него помоћу дигитрона тј. калкулатора). Једино је важно да умеш исправно да запишеш израз којим се тражено израчунавање изражава. У математици би се одговарајући израз записао као (874+437)·(915-364). Ако се сетиш да се множење изражава знаком * и ако ти кажемо да у програмском језику Python можеш употребљавати заграде на исти начин као у математици, онда ти је јасно да претходни математички задатак можеш лако решити тако што употребиш израз (874 + 437) * (915 - 364).

У програмском окружењу на свом рачунару откуцај овај израз и прочитај вредност коју ће ти Python дати.

Заграде су у претходном сложеном изразу неопходне, јер је приоритет операција потпуно исти као у математици (прво се израчунава оно што је у заградама, затим множење и дељење, па тек онда сабирање и одузимање).



Питање 1.
Која је вредност израза 5 + 5 * 5?
(А) 15
(Б) 30
(Ц) 50
(Д) 125

Одговор: Б

Питање 2.
Која је вредност израза (5 + 5) * 5?
(А) 15
(Б) 30
(Ц) 50
(Д) 125

Одговор: Ц

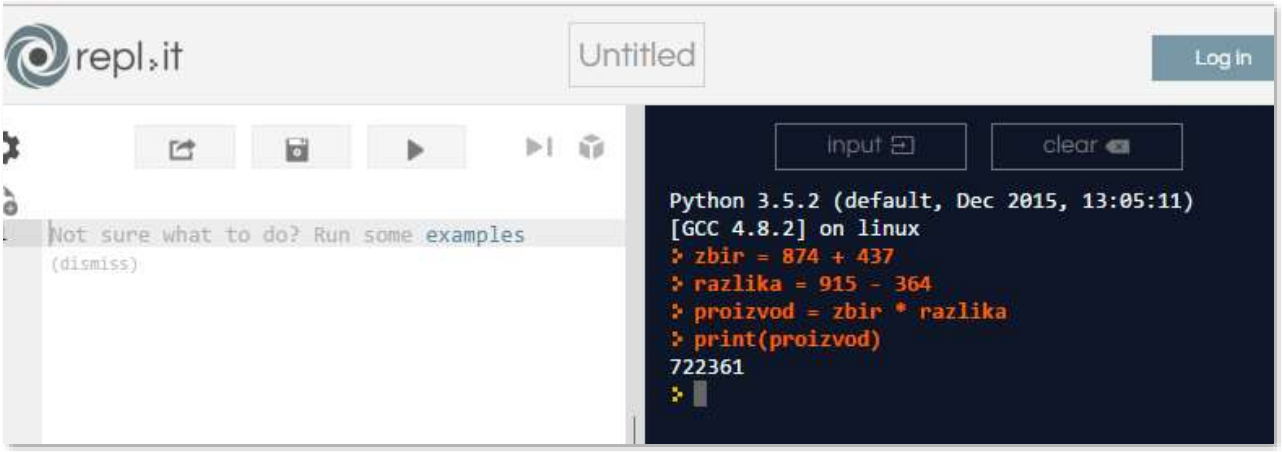
Промењиве - имена међурезултата

Писање сложених израза се може избећи, а програм се може начинити мало разумљивијим ако међурезултате именујемо. Погледај видео лекцију

Python - аритметичке операције

Погледајмо наредни пример програма који такође решава претходни задатак:

Израчунај производ збира бројева 874 и 437 и разлике бројева 915 и 364.





Збиру смо доделили име `zbir`, разлици име `razlika`, а производу име `proizvod` (уместо zbir и razlika могли смо, на пример, користити и имена `prvi_cinilac`, `drugi_cinilac`). Иако се на овај начин добија програм који мало дужи него полазни, он је мало разумљивији, јер се његовим читањем може јасно видети да се прво тражи израчунавање збира, затим разлике и затим њиховог производа. То се десило пре свега захваљујући пажљивом одабиру имена која смо употребили и веома је важно у програмима користити имена која читаоцу програма дају назнаке шта тај програм заправо израчунава. Имај на уму да програме чита рачунар који их извршава и њему је сасвим свеједно која су имена употребљена, али и људи који те програме пишу, исправљају и дорађују, а њима је прилично важно да текст програма лако разумеју. У већини случајева си читалац програма управо ти, тако да давањем илустративних имена променљивима данас помажеш заправо себи у будућности.

Имена која смо дали међурезултатима се у програмирању називају **променљиве**. Променљиве су јако важан концепт о коме ће бити много више речи касније. До тада ћемо их користити на потпуно исти начин у математици - само као имена придружена одређеним вредностима.

Рецимо да постоје и нека правила која се односе на имена (каже се и идентификаторе) која можемо користити. Прво, постоји разлика између великих и малих слова и није исто да ли смо употребили zbir или Zbir. Препоручује се да у именима користимо само слова енглеске абецеде (тзв. ошишану латиницу), бројеве и подвлака (симбол _) који ћемо користити да повежемо више речи у једно име. У именима не можемо користити размаке, зарезе и слично, нити име можемо започети цифром. Дозвољена имена су, на пример, x, obim, drugi_sabirak, broj_sekundi, a2, а недозвољена су, на пример, 3d_grafika (јер почиње цифром), prvi sabirak (јер садржи размак) и jezik_c# (јер садржи недозвољени знак #).

Задатак 2.
Израчунај разлику производа и збира бројева 874 и 437.

Покушај самостално да решиш овај задатак. Помоћ и решење потражи у интерактивном уџбенику.