

3.1.

Основе изабраног програмског језика

Кључне речи

Програмски језик, Пајтон, синтакса, развојно окружење.



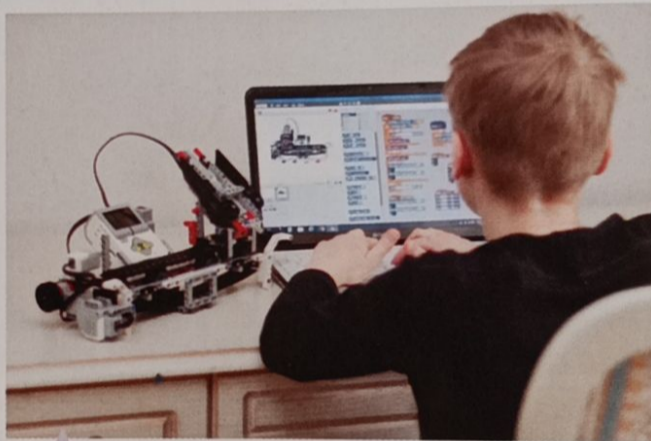
Подсети се:

- ▶ шта је алгоритам;
- ▶ објасни разлику између податка и информације;
- ▶ како се програмира у Скрачу.

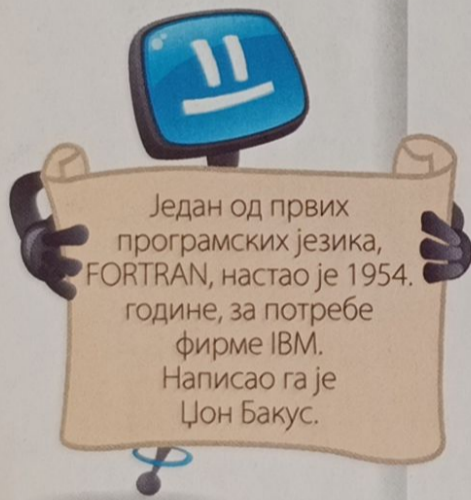
Размисли

Пошто си савладао/савладала основе програмирања, подсети се из колико се етапа састоји процес програмирања. Наведи их и објасни.

У 5. разреду си на примеру Скрача (енг. Scratch) научио/научила да програмирање може бити занимљиво и креативно (сл. 3.1). Користећи готове блокове на брз и једноставан начин си креирао/креирала програм.



3.1. Програмирање у Скрачу



Већ ти је познато да је за програмирање потребно да разликујеш податак од информације, као и да направиш редослед радњи за решавање проблема у облику алгоритма. На тај начин ћеш моћи рачунару да даш добру наредбу за извршавање програма.

Наредба је најмањи самостални елемент програмског језика и описује акцију која треба да буде извршена. Наредбе могу бити једноставне и сложене.

Сазнај више

Подела програмских језика

По степену зависности програмског језика од рачунара, програмске језике делимо на:

- машински зависне (машински и симболички језик);
- машински независне (језици вишег нивоа).

Пајтон спада у језике вишег нивоа.

Програмски језици могу бити визуелни и текстуални. Скрач је пример визуелног програмског језика и добар је начин да се млађи ученици упознају са основама програмирања. Ове године научићеш да пишеш програме у текстуалном програмском језику који се зове **Пајтон** (енгл. Python). Пајтон је добар за почетнике у програмирању јер је једноставан за рад и брзо се савладава.

Разлика у програмирању између Скрача, као визуелног програмског језика и Пајтона, као текстуалног програмског језика приказана је у наставку.

Оно што бисмо у писању приказали на следећи начин:

Реци ми дужину правоугаоника.

Упамтићу дужину.

Реци ми ширину правоугаоника.

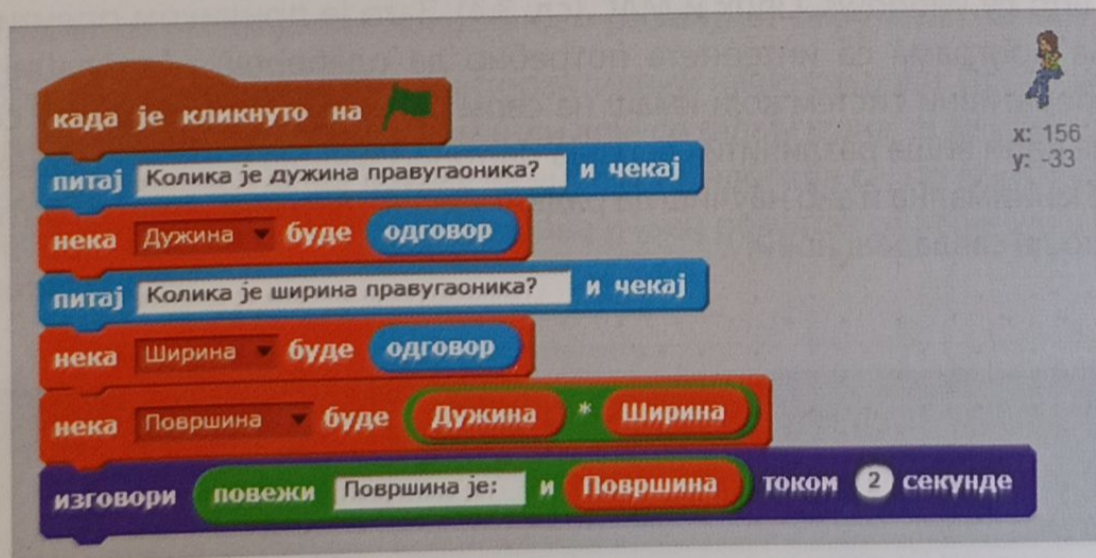
Упамтићу ширину.

Рачунам површину по формули:

површина = дужина · ширина

Површина правоугаоника износи:...

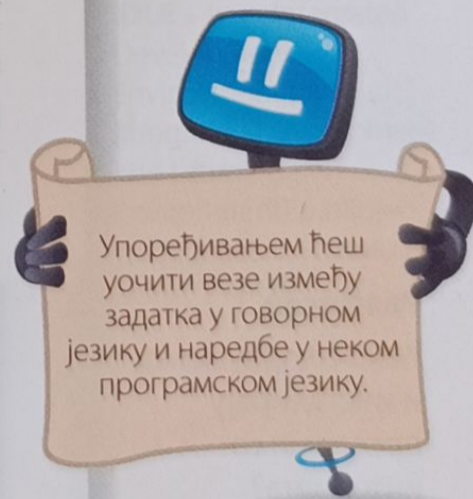
у Скрачу изгледа као на слици 3.2.



3.2. Изглед наредбе у Scratch-у

Исти налог се може извршити и у Пајтону, куцањем следеће команде:

```
a = int(input("Unesi duzinu pravougaonika: "))  
b = int(input("Unesi sirinu pravougaonika: "))  
p = a * b  
print ("Povrsina pravougaonika iznosi: ", p)
```



Основе Пајтона

Пајтон је врста текстуалног програмског језика. Једноставно се инсталира на рачунару и брзо се учи. Може да се користи за писање једноставнијих програма, а може и за неке сложеније. Број функција у овом програмском језику није велики, па ти не треба много времена за израду нових програма. Такође, Пајтон садржи отворени код који је бесплатан за коришћење. То значи да програмери могу сами да пишу кодове и креирају библиотеке, али те исте библиотеке могу да користе и други програмери.

Као и остали програмски језици, Пајтон има своје симболе, односно низ правила која дефинишу правилну језичку конструкцију. То представља Пајтонову **синтаксу** (енг. *Syntax*). У случају да се приликом писања програма нису поштовала правила о писању програмских конструкција, јавља се **синтаксна грешка** (енг. *Syntax Error*).

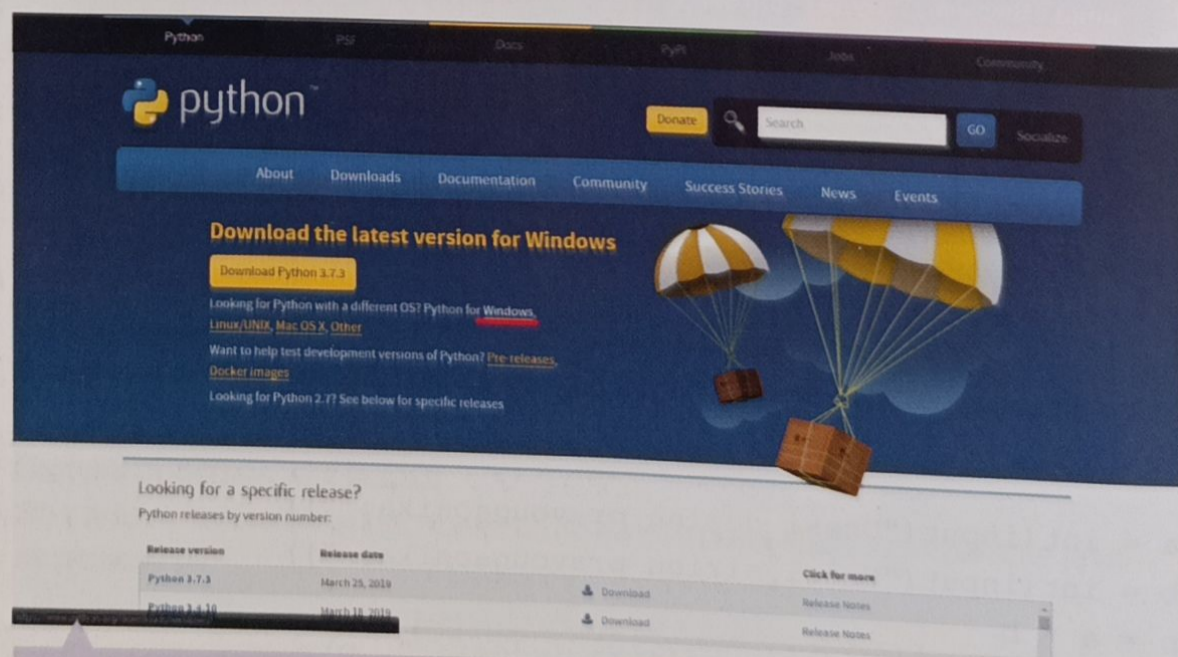
Пајтонова синтакса је читљива и једноставна. Нема компликованих корака и лош унос података неће изазвати грешку у самом коду, па програмери имају више времена да се баве проблемом који треба да реше, него дешифровањем програмског кода.

Инсталација програмског језика Пајтона

Пајтон је бесплатан софтвер и може се преузети са сајта <https://www.python.org/downloads/>. Подржава га већина оперативних система, као што су Windows, Linux и MAC (сл. 3.4). Зато је приликом преузимања програма са интернета потребно да одабереш инсталацију за оперативни систем који имаш на свом рачунару. На интернету се може наћи више различитих верзија Пајтона. Разлика између софтвера је минимална и ако научиш да радиш у једној верзији, неће ти бити тешко да савладаш новију.

Размисли

Како се преузимају и инсталирају програми са интернета?



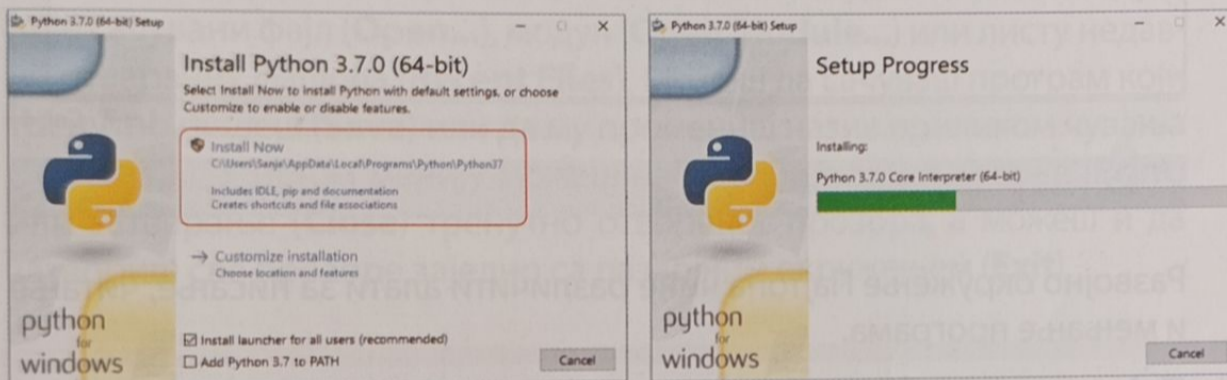
Looking for a specific release?
Python releases by version number:

Release version	Release date	Download	Click for more
Python 3.7.3	March 25, 2019	Download	Release Notes
Python 3.6.8	March 18, 2019	Download	Release Notes

3.4. Одабир одговарајуће инсталације у зависности од оперативног система који користи ученик

Када покренеш преузимање програма, појавиће се питање где желиш да инсталациони фајл буде сачуван. У зависности од подешавања на веб-прегледачу, преузимање се може започети и без овог питања, па ће се у том случају фајл сачувати у фолдеру Downloads. Дуплим кликом на фајл отвараш нови прозор у коме треба да одабереш поље **Install Now** и инсталација почиње (сл. 3.5).

Када је инсталиран, програм се покреће помоћу наредбе: **Start** → **All programs** → **Python** → **IDLE (Python 3.7 64-bit)**.

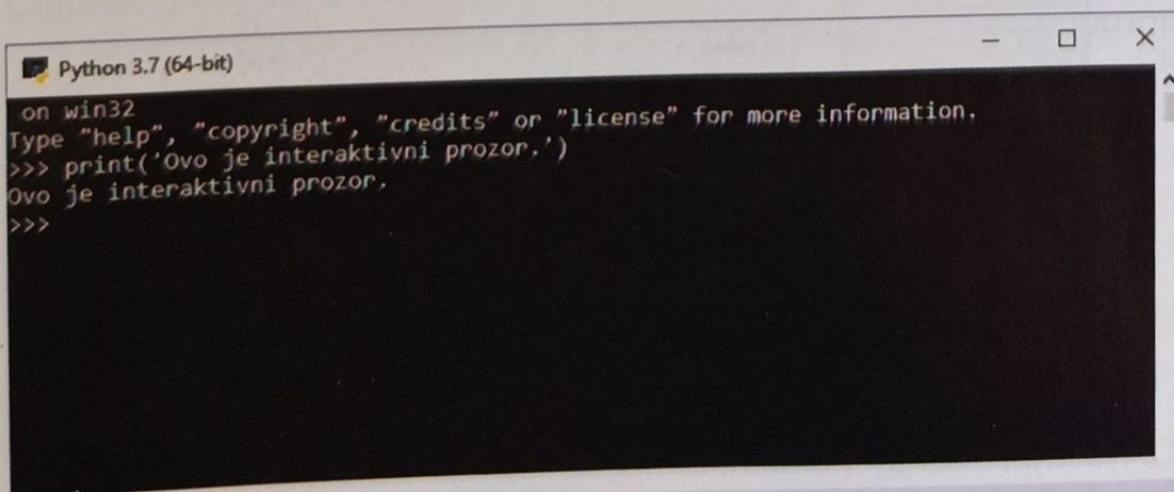


3.5. Почетак инсталације Пajтона

Радно окружење Пajтона

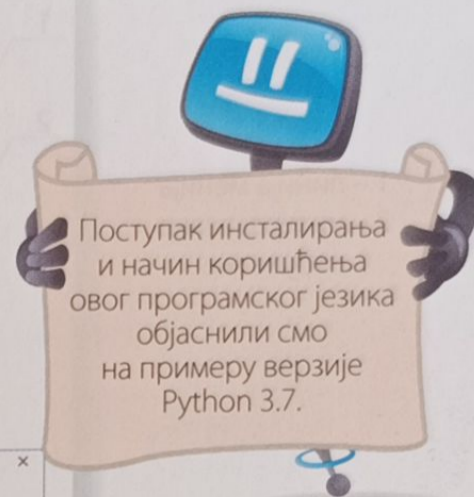
Пajтон се може изводити у различитим окружењима. Од тренутка када га инсталираш на свом рачунару, можеш му приступити на два начина: покретањем интерактивног окружења **Python interpreter** преко Start менија или покретањем **развојног окружења IDLE***.

Програм се најбоље пише интерактивним начином рада. Програмски код исписује се у облику команди које се задају једна за другом. Символ `>>>` представља линију за унос команде која се, по притиску на тастер Enter на тастатури извршава и њен резултат се исписује у следећем реду.



3.6. Интерактивно окружење Пajтона

IDLE је развојно окружење које се добија са инсталацијом програмског језика Пajтон. Након покретања, појављује се интерактивни прозор **Python Shell** (сл. 3.7). То је интерактивно окружење у коме пишеш програмски код, а можеш у исто време да видиш и резултат.

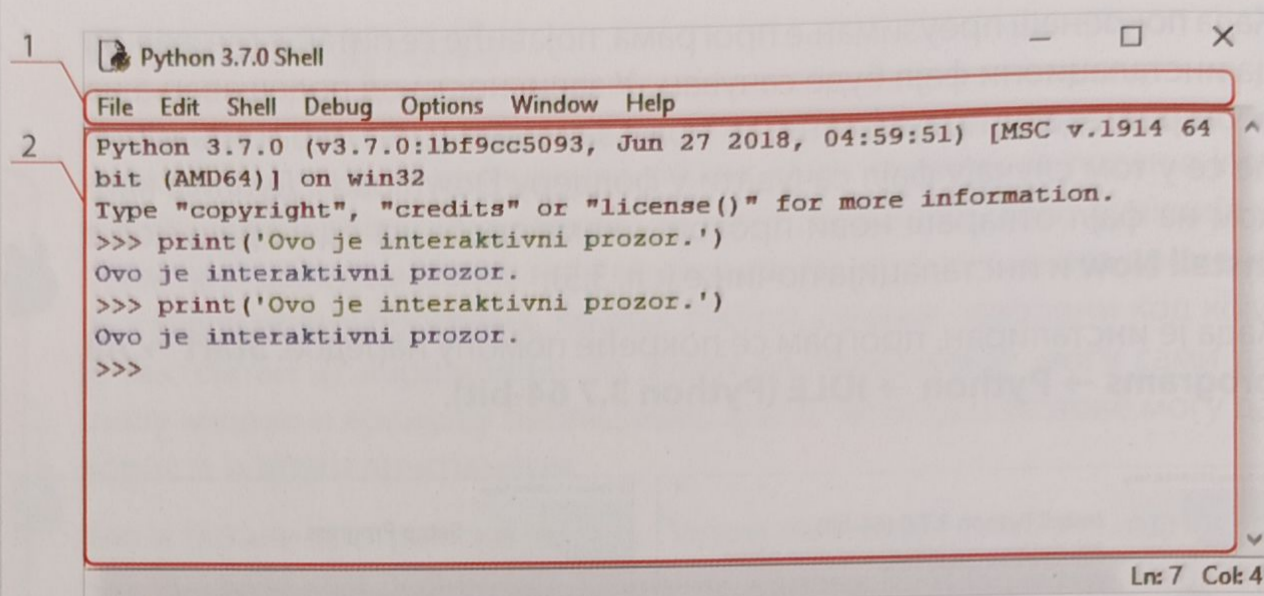


Поступак инсталирања и начин коришћења овог програмског језика објаснили смо на примеру верзије Python 3.7.

*

IDLE – (енг. Integrated Development Environment или Integrated Development and Learning Environment) развојно окружење програмског језика Пajтон.

- 1 – линија менија
- 2 – интерактивно окружење

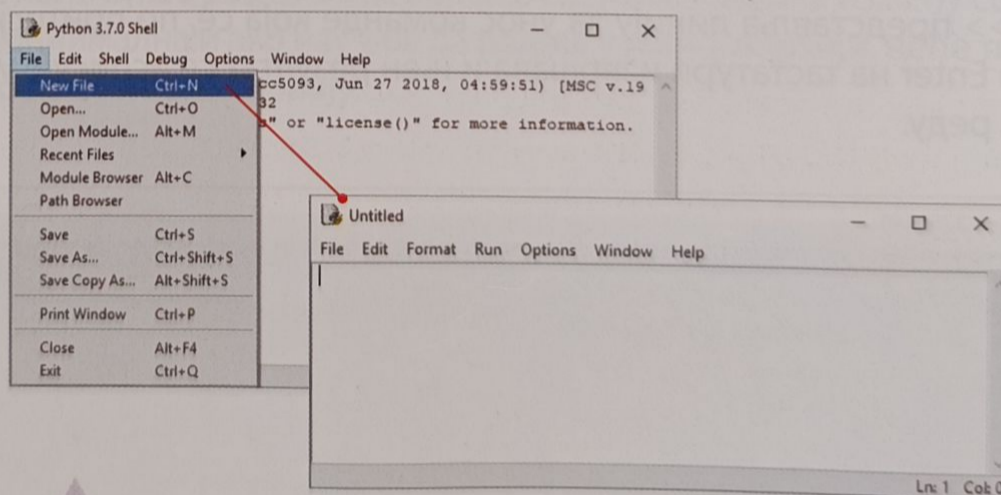


3.7. Развојно окружење Пајтона

Развојно окружење Пајтона чине различити алати за писање, читање и мењање програма.

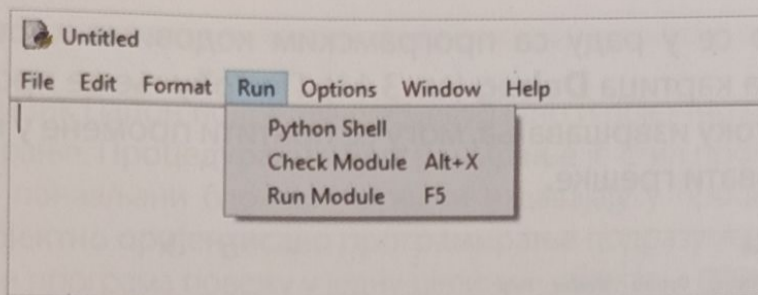
Пајтон конзола је програм који ти омогућава да унесеш једну линију програмског кода, коју можеш да понављаш. Међутим, у раду са интерактивним прозором, команде које пишеш се не чувају. Да би се програм реализовао, команде се морају запамтити и поновити увек у истом редоследу. Када желиш да извршиш сложенију наредбу више пута, треба да сачуваш цео код. За то се користи нови прозор, **Untitled**.

Нови позор се отвара преко главног менија, помоћу опције **New File** на картици **File** (сл. 3.8) или пречицом **CTRL + N**. Ту можеш откуцати команду коју ћеш касније понављати, а сачуваћеш је помоћу наредбе **File** → **Save** или пречице **CTRL + S**.



3.8. Отварање новог прозора у Пајтону

Сачувану команду покрећеш преко менија **Run** (сл. 3.9). Избором опције **Python Shell** команда се приказује у већ отвореном интерактивном окружењу. Ако желиш да провериш команду коју си откуцао/откуцала, пре него што је примениш у програму на коме радиш, бираш опцију **Check Module**. Ако је команда добро написана, можеш поново да покренеш интерактивно окружење Python Shell, а затим да извршиш команду (**Run Module**).



3.9. Мени Run

Осим ове могућности, преко картице File можеш да отвориш неки раније сачувани фајл (**Open...**), модул (**Open Module...**) или листу недавно отворених фајлова (**Recent Files**). Можеш да сачуваш програм који тренутно пишеш (**Save**) или да му промениш назив приликом чувања (**Save as...**). У истом менију можеш наћи опције за штампање (**Print**) или затварање (**Close**) тренутно отвореног прозора, а можеш и да затвориш све прозоре заједно са развојним окружењем (**Exit**).

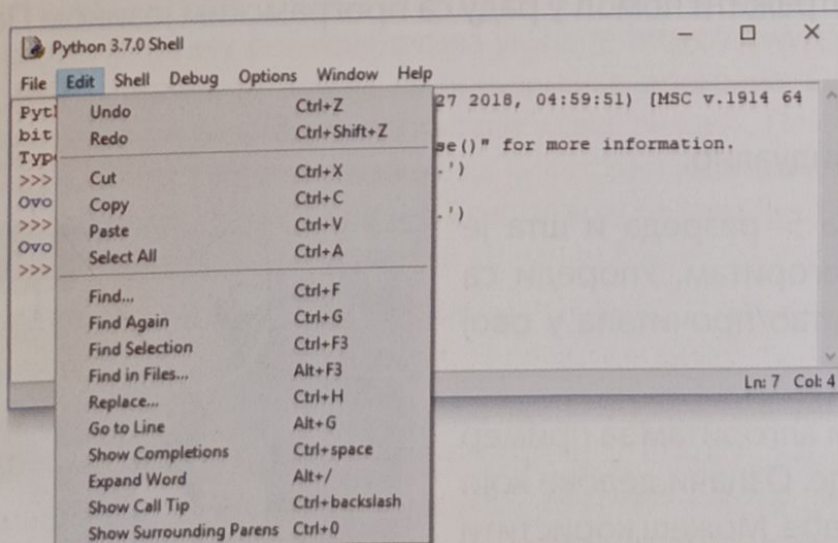
Током програмирања користе се два прозора у исто време. Важно је подесити њихову величину, да би могао/могла да пратиш промене до којих долази куцањем одређених наредби.

Важно!

За рад у интерактивном окружењу, писање и мењање програма, користи се мени **Edit** (сл. 3.10). Он даје могућност за поништавање (**Undo**) или понављање (**Redo**) последње промене у тренутно отвореном прозору, исецање и премештање (**Cut**) или копирање (**Copy**) одређеног садржаја у системски **клипборд*** и његово додавање (**Paste**) у тренутно отворени прозор. Осим поменутих, преко овог менија можеш отворити прозор за претрагу садржаја (**Find...**), поновити последњу претрагу ако постоји (**Find Again**), пронаћи обележени садржај тренутно отвореног прозора (**Find Selection**), отворити прозор за претрагу са захтевом да се резултати издвоје у нови прозор (**Find in Files...**). Обележени део кода можеш да мењаш помоћу опције **Replace**.

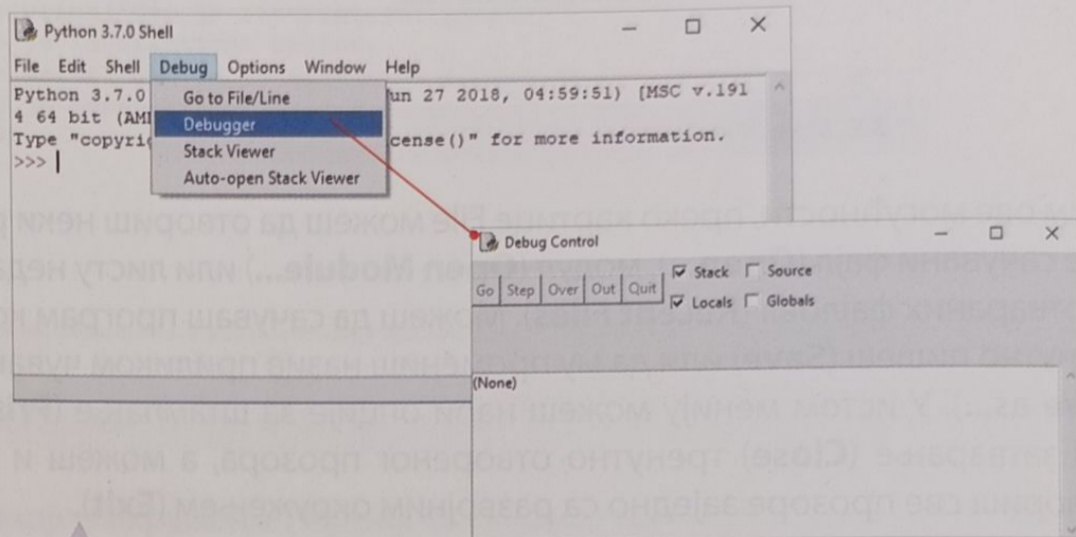
*

Клипборд (енг. **Clipboard**) – јесте део меморије у који се привремено смешта исечени или копирани текст или графички објекат.



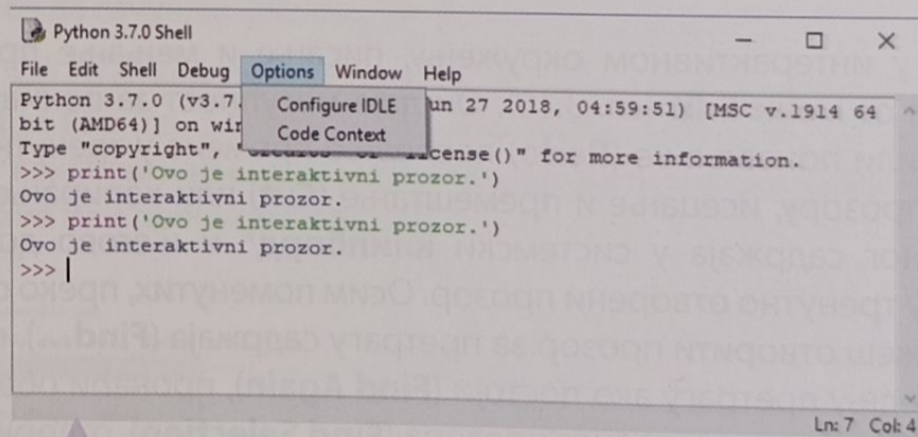
3.10. Мени Edit

Веома често се у раду са програмским кодовима користи алатка **Debugger** на картица **Debug** (сл. 3.11). Помоћу ње се проверава рад програма у току извршавања, могу се пратити промене у коду и на тај начин откривати грешке.



3.11. Алатка „Дебагер“

Мени **Options** (сл. 3.12) нуди могућност отварања прозора за конфигурацију и промену поставки за фонтове, украсе, теме, додатне изворе помоћи и слично (**Configure IDLE**), али служи и за премештање (**Code Context**).



3.12. Картица „Options“

Мени **Window** служи за рад са прозорима, док помоћу менија **Help** можеш потражити помоћ у раду са програмским језиком Пајтон.

ЗАДАТАК

1. Задатак ради индивидуално.
2. Подсети се градива 5. разреда и шта је сценарио, а шта алгоритам. Упореди са оним што си прочитао/прочитала у овој лекцији.
3. Напиши сценарио и алгоритам за пример прављења лимунаде. Означи делове који представљају наредбе. Можеш користити свеску и уџбеник из петог разреда.



Програмски језик Пајтон подржава процедурално и објектно оријентисано програмирање. **Процедурално програмирање** је стил програмирања где се често понављани блокови наредби издавајају у процедуре (или функције). **Објектно оријентисано програмирање** подразумева да се подаци и делови програма повежу у једну целину – објекат.

**Сазнај
више**

Програмски језици могу бити визуелни и текстуални програмски језици.

Пајтон је текстуални програмски. Једноставно се инсталира и брзо се учи. Подржава га више оперативних система (Windows, Linux и MAC OS) и за сваки од њих постоји посебна инсталација. Уз инсталацију се добија **развојно окружење – IDLE**.

Радно окружење у Пајтону састоји се од линије менија и радне површине у којој пишемо код програма.

Запамти!

1. Како се деле програмски језици?
2. Наведи разлике и сличности приликом програмирања у визуелним и текстуалним програмским језицима.
3. Објасни поступак инсталирања програмског језика Пајтон.
4. Од чега се састоји IDLE?

**Провери
знање**

1. Поделите се у групе од по три до четири ученика. Нека свака група одабере задатак А или задатак Б.
2. Прочитајте текст задатка који сте одабрали и урадите га.
3. На једном од наредних часова формирајте нове групе, тако да свака група има и чланове група које су радиле задатак А и чланове група које су радиле задатак Б. Размените знања и искуства везана за резултате рада на задацима А и Б.

Истражи

Задатак А

Верзије Пајтона за преузимање са интернета стално се мењају и допуњавају. У адресну линију веб-прегледача укуцајте <https://www.python.org/downloads/> и проучите опције за преузимање програма. Опишите их и упоредите са оним што сте прочитали у овој лекцији. Покушајте да у опис укључите одговоре на питања: да ли постоји новија верзија, да ли постоји допуна програма, да ли је важно за коју се верзију оперативног система преузима програм и на који се начин бирају опције и покреће преузимање.

Задатак Б

Покрените Пајтон и проучите главни мени. Својим речима опишите опције које нуди картица Edit. Упоредите опције ове картице са оним опцијама које се могу наћи у програмима као што су Microsoft Word или Microsoft Power Point. Опишите сличности и разлике. Затим, проучите опције картице Option. Консултујте се са наставником/наставницом и опишите опције ове картице.

3.2.

Основне аритметичке операције и типови података

Кључне речи

Изрази, аритметички оператори, логички оператори, константа, променљива.



Подсети се:

- ▶ основних рачунских операција;
- ▶ који бројеви припадају скупу целих бројева, а који скупу реалних бројева;
- ▶ Еуклидовог алгоритма.

Од најранијег детињства сусретао/сусретала си се са основним аритметичким операцијама: сабирањем, одузимањем, множењем и дељењем. Можеш их користити и у програмском језику Пајтон, а то ћемо ти показати на примеру бројева 8 и 2.

Рачунске операције

Рачунска операција	Математика	Програмски језик Пајтон
Сабирање	$8 + 2$	$8 + 2$
Одузимање	$8 - 2$	$8 - 2$
Множење	$8 \cdot 2$	$8 * 2$
Дељење	$8 : 2$	$8 / 2$

Записи, попут $8 + 2$, $8 - 2$, $8 * 2$ и $8 / 2$, називају се **изрази**. Символи који се користе за означавање аритметичких операција називају се **аритметички оператори**, док се симболи за логичке операције називају **логички оператори**.

Размисли

Како се израчунава збир у Скрачу?

Који аритметички оператори се користе у Пајтону?

$+$, $-$, $*$ и $/$.

Да ли се користе још неки симболи?

Да, за целобројно дељење $//$, а за остатак при дељењу $\%$.



Важно!

Скуп вредности које имају заједничке особине и скуп операција које се могу применити на те вредности назива се тип података у програмским језицима.

1. Вежба се ради индивидуално/у пару.
2. Покрени програмско окружење Пајтона.
3. Откуцај наведене изразе: $8 + 2$, $8 - 2$, $8 * 2$ и $8 : 2$.
4. Притисни тастер Enter да би проверио/проверила које ће ти вредности аритметичких израза окружење дати. Ако си све унео/унела како треба, добићеш резултате приказане на слици 3.13.
5. Покрени интерактивно окружење Пајтона.
6. Откуцај исте записе, али примени команду:
`print(8+2),print(8-2),print(8*2),print(8/2),`
 па упореди резултате са вредностима које си раније добио/добила.

```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27
64 bit (AMD64)] on win32
Type "copyright", "credits" or "license
>>> 8 + 2
10
>>> 8 - 2
6
>>> 8 * 2
16
>>> 8 / 2
4.0
>>> |
```

```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27
64 bit (AMD64)] on win32
Type "copyright", "credits" or "license
>>> print(8 + 2)
10
>>> print(8 - 2)
6
>>> print(8 * 2)
16
>>> print(8 / 2)
4.0
>>> |
```

3.13. Приказ основних аритметичких операција

Напомена: Бројеви 8 и 2 припадају скупу целих бројева. Команда `print()` коју смо користили извршила се над типом података „цео број“.

Без обзира да ли куцаш размак између броја и оператора, добићеш исти резултат. Размаци нису неопходни, али је препорука да се пишу због прегледности програма и читљивости програмског кода.

Важно!

Реално и целобројно дељење

У већини програмских језика постоје два облика дељења: **реално и целобројно дељење**. Резултат реалног дељења припада скупу реалних бројева. На пример, када 7 поделиш са 2, добијеш количник 3,5. Код целобројног дељења количник је увек цео број. Значи, када поделиш 7 са 2, добијеш количник 3, а остатак је 1.

Размисли

Како се врши дељење броја са остатком?

Аритметички оператор за реално дељење је /, а за целобројно дељење //. На слици 3.14 дат је пример реалног и целобројног дељења.

```
>>> 7 / 2
3.5
>>> 7 // 2
3
```

3.14. Приказ реалног и целобројног дељења

Приликом целобројног дељења појављује се остатак при дељењу (тзв. модул). Он се у Пајтону израчунава коришћењем аритметичког оператора %. На пример, када у Пајтону желимо да израчунамо количник броја 35 и 10, можемо записати следеће (сл. 3.15):



```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1
914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 35 / 10
3.5
>>> 35 // 10
3
>>> 35 % 10
5
>>>
```

Реално дељење	35 / 10 = 3.5 количник је реалан број
Целобројно дељење	35 // 10 = 3 количник је 3, остатак је 5
Дељење са остатком	35 % 10 = 5 количник је 3, остатак је 5

3.15 Приказ реалног, целобројног и дељења са остатком

Сложени изрази, приоритет рачунских операција и заграде

У Пајтону се могу решавати и задаци који укључују више рачунских операција и тада користиш сложеније изразе. Да би исправно записао/записала израз, треба да водиш рачуна о приоритету оператора, па можеш користити заграде. Заграде се у програмском језику Пајтон користе на исти начин као у математици (сл. 3.15). Прво се израчунава оно што је у заградама, онда множење и на крају дељење.

Пошто је приоритет операција исти као у математици, заграде су неопходне.



```
>>> (35 - 2) * (19 + 6) / 3
275.0
>>>
```

3.16. Приказ сложенијег изрази у Пајтону

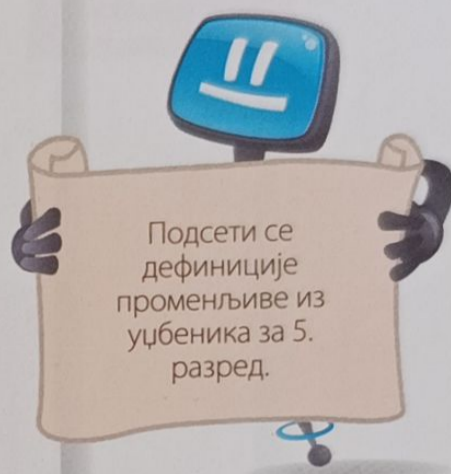
Променљиве

Научио/научила си у 5. разреду да променљива нема сталну вредност већ да се њена вредност уноси (тражи се од корисника рачунара да је унесе преко тастатуре) или добија током извршавања програма. У програмирању ти је понекад потребна променљива чија се вредност не мења. Ове променљиве називају се **константе**. На пример, ако желиш да израчунаш како сила гравитације делује на тела, уместо уноса ове силе, која увек износи $9,81 \text{ m/s}^2$, можеш употребити константу g , којом се ова сила означава.

Веома је важно да се у програму користи име податка сачувано у меморији рачунара (на пример: висина_зида = 10). Те податке морамо именовати тако да читалац програма зна тачно у ком делу програма може да их употреби.

Назив којим смо именовали податке, не можемо да мењамо, али вредност тих података може се мењати у току извршавања програма и зато их и називамо **променљивима** (енг. *variable*). Међутим, можемо им на почетку програма доделити сталне вредности, као у математици, па ће у том случају остати непромењене до краја извршавања програма.

Употребом променљивих рад у програму постаје лакши. Ти можеш да мењаш програм који пишеш како желиш, али ћеш на основу имена променљиве увек знати шта она представља.



Подсети се дефиниције променљиве из уџбеника за 5. разред.

Размисли

Која је разлика између променљиве у математици и променљиве у програмирању?

ВЕЖБА

1. Вежба се ради индивидуално/у пару.
2. На слици 3.17 приказано је како се у Пајтону записује променљива `zbir`. Проучи приказ.
3. Дефиниши променљиву чије је име `zbir`. Без покретања програмског окружења одреди вредности израза $28 + 7$, а затим провери резултат у Пајтону.
4. Променљиве су нам потребне и код именовања међурекултата израчунавања. Проучи како се у Пајтону збиру бројева -6 и -18 додаје разлика бројева -9 и -7 (сл. 3.18).

```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1
914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> zbir = 10 + 15
>>> print(zbir)
25
>>> |
```

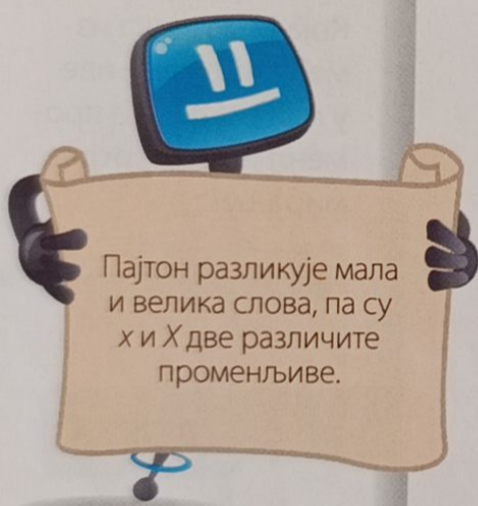
3.17. Приказ променљиве `zbir`

```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1
914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> zbir = -6 + (-18)
>>> razlika = -9 - (-7)
>>> rezultat = zbir + razlika
>>> print(rezultat)
-26
>>>
```

3.18. Приказ именовања међурекултата израчунавања

Код именовања променљивих потребно је придржавати се одређених правила.

Променљива:	Дозвољено	Недозвољено
може садржати слова бројеве и доњу црту;	✓	
може почети бројем;		✓
може садржати размаке, цртице и остале знаке интерпункције;		✓
може садржати кључне речи које су резервисани називи (као што су: <i>True, False...</i>);		✓
може садржати слова попут <i>č, ć, ž, š, đ, Č, Ć, Ž, Š, Đ.</i>	✓	



Променљиве служе и за учитавање улазних података. На пример, површину правоугаоника можемо израчунати ако се зна да су му странице 4,5 cm и 6,5 cm (сл. 3.19). Обично се странице правоугаоника обележавају словима *a* и *b*, па је тако и у овом примеру.

```
>>> a = 4.5
>>> b = 6.5
>>> P = a * b
>>> print(P)
29.25
```

3.19. Приказ учитавања улазних података

Програм за израчунавање површине правоугаоника за било које унете вредности можеш написати и тако што ће се од корисника тражити да унесе вредности за израчунавање.

ЗАДАТАК

Користећи раније стечено знање из математике и оно што си научио/научила у овој лекцији, напиши програм у којем ћеш израчунати:

- збир бројева 72 и 64;
- разлику бројева 135 и 68;
- производ бројева 13 и 99;
- количник бројева 168 и 90;
- количник збира бројева 14 и 91 и разлике бројева 18 и 9;
- збир производа бројева 2 и 7 и количника бројева 96 и 8.

У програмирању је важно писати коментаре, и то са хаштагом (#). Они се не извршавају, али су ту као помоћ осталим програмерима да лакше разумеју твој програмски код. Коментари су важан део програмског кода (документација).

Сазнај
више

```
>>> # Ово је коментар.  
>>> # Коментар у једној линији  
>>> ""  
    Коментар у више линија  
    ""
```

3.20. Коментар у Пајтону

Основне аритметичке операције (сабирање, одузимање, множење и дељење) можеш да користиш и у програмском језику Пајтон. Записи, попут $8 + 2$, $8 - 2$, $8 * 2$ и $8 / 2$, називају се **изрази**. Символи који се користе за означавање аритметичких операција називају се **аритметички оператори**, док се симболи за логичке операције називају **логички оператори**.

Запамти!

У већини програмских језика постоје два облика дељења: **реално и целобројно дељење**. Резултат реалног дељења припада скупу реалних бројева, док код целобројног дељења количник је увек цео број. Аритметички оператор за реално дељење је $/$, а за целобројно дељење $//$. У Пајтону се могу решавати и задаци који укључују више рачунских операција и тада користиш сложеније изразе. Да би исправно записао/записала израз, треба да водиш рачуна о приоритету оператора, па можеш користити заграде.

Веома је важно да се у програму користи име податка сачувано у меморији рачунара. Те податке морамо именовати тако да корисник зна тачно у ком делу програма може да их употреби. Назив којим смо именовали податке, не можемо да мењамо, али вредност тих података може се мењати у току извршавања програма и зато их и називамо **променљивима**.

1. Како се у Пајтону записују основне аритметичке операције?
2. Шта су променљиве и како се именују?
3. Без покретања програмског окружења Пајтон одреди вредности израза $125 / 5$, $75 // 10$ и $15 \% 3$. Резултат провери у Пајтону.

Провери
знање

Поделите се у групе од по три до четири ученика. Свака група ученика треба (уз помоћ наставника) да осмисли по један проблем који се може решити на основу до сада пређеног градива, а који се може решити употребом наредби које су поменуте у лекцији. На неком од часова утврђивања групе треба да презентују свој проблем и поступак његовог решавања у Пајтону.

Истражи

https://petlja.org/biblioteka/r/lekcije/python-zbirka-pitanja/deljenje_pitanja

Корисни
линкови

3.3.

Функције

Кључне речи

Функција, уграђене функције, минимум, максимум.



Подсети се:

- ▶ како се у програмском окружењу Пајтон извршава рачунање сложенијих израза;
- ▶ која је сврха коришћења наредбе `print()`.

Функције у програмском окружењу Пајтон имају широку намену. Корисне су за дељење великих програма на мање делове, да би се учинили краћим, а због тога и читљивијим. Могу се извршавати више пута у зависности од задатка тј. проблема. Позивањем функција извршава се низ наредби.

У Пајтону се најпре дефинише функција која израчунава површину правоугаоника. Функције се дефинишу наредбом `def`, именом и улазним параметрима. Наредба се завршава двотачком, а код који је део функције пише се у блоку испод наредбе `def`.

```
>>> def print_povrsina ():
```

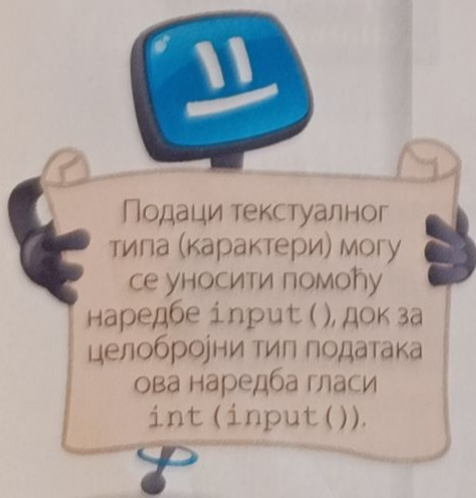
Пајтон има неке већ написане функције за команде које се често користе. То су, **уграђене функције** (енг. *Built-in*). Оне омогућавају много лакше и брже писање програма.

Функција `print` ти је већ позната, јер си је до сада користио/користила за исписивање бројева. Исту функцију можеш користити за исписивање текста, тј. низа карактера који се налазе између знакова навода.

```
>>> print ("Unesi vrednost prve stranice.")
Unesi vrednost prve stranice.
```

Осим ње, често се користи функција `input()`. Позивањем ове функције можеш да унесеш неку улазну вредност у програм. На пример:

```
>>> grad = input('Unesi ime grada: ')
Unesi ime grada: Beograd
>>> print('Glavni grad, ', grad)
Glavni grad, Beograd
```

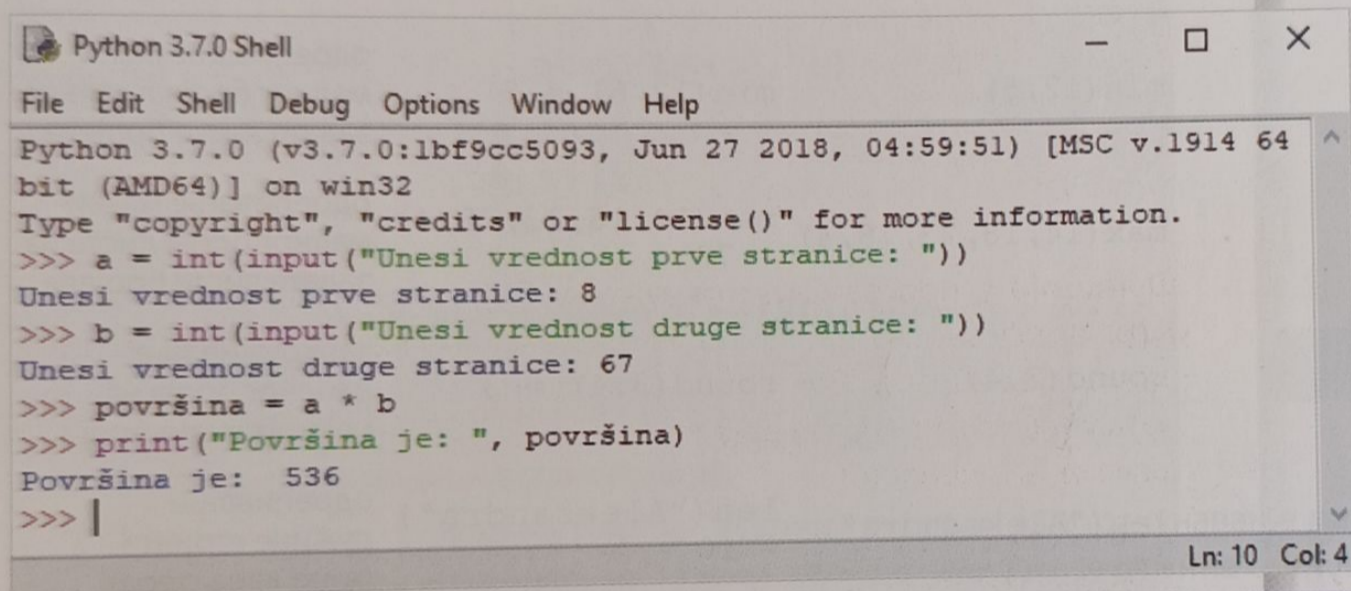


Подаци текстуалног типа (карактери) могу се уносити помоћу наредбе `input()`, док за целобројни тип података ова наредба гласи `int(input())`.

Да би унели целе бројеве користимо функцију `input()`, такође морамо дефинисати да желимо да и наш програм разуме да је то цео број корићењем `int`-а, који се пише `int`.

```
>>> a = int(input("Unesi vrednost prve stranice: ")).
```

Када све ово научиш, можеш написати програм за израчунавање површине правоугаоника (сл. 3.21).



```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64
bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> a = int(input("Unesi vrednost prve stranice: "))
Unesi vrednost prve stranice: 8
>>> b = int(input("Unesi vrednost druge stranice: "))
Unesi vrednost druge stranice: 67
>>> površina = a * b
>>> print("Površina je: ", površina)
Površina je: 536
>>> |
```

3.21. Приказ уноса улазних променљивих

ЗАДАТАК

1. Задатак се ради индивидуално/у пару.
2. У наставку су дати делови програма за израчунавање просека оцена.
3. Правилно их допуни и израчунај у Пајтону просек својих оцена.

```
print ('Ocena iz srpskog jezika i književnosti: ')
s = input()
print ('Ocena iz matematike: ')
m = input()
print ('Ocena iz informatike i računarstva: ')
i = input()
print ('Ocena iz biologije: ')
b = input()
print ('Ocena iz geografije: ')
g = input()
zbir_ocena =
prosek =
print ('Prosek ocena je: ', prosek)
```