

INFORMATIČKI KLUB

FUTURA

LIGA PROGRAMIRANJA



LIGA PROGRAMIRANJA U PYTHONU ZA OSNOVNE ŠKOLE – 2. RADIONICA

Mario Miličević, Informatički klub FUTURA
Dubrovnik, 8. studenog 2014.



Zaštićeno licencom <http://creativecommons.org/licenses/by-nc-sa/3.0/hr/>



Creative Commons



- ☐ **slobodno smijete:**
 - **dijeliti** — umnožavati, distribuirati i javnosti priopćavati djelo
 - **remiksirati** — prerađivati djelo
- ☐ **pod slijedećim uvjetima:**
 - **imenovanje.** Morate priznati i označiti autorstvo djela na način kako je specificirao autor ili davatelj licence (ali ne način koji bi sugerirao da Vi ili Vaše korištenje njegova djela imate njegovu izravnu podršku).
 - **nekomercijalno.** Ovo djelo ne smijete koristiti u komercijalne svrhe.
 - **dijeli pod istim uvjetima.** Ako ovo djelo izmijenite, preoblikujete ili stvarate koristeći ga, preradu možete distribuirati samo pod licencom koja je ista ili slična ovoj.

U slučaju daljnjeg korištenja ili distribuiranja morate drugima jasno dati do znanja licencne uvjete ovog djela. Najbolji način da to učinite je linkom na ovu internetsku stranicu.

Od svakog od gornjih uvjeta moguće je odstupiti, ako dobijete dopuštenje nositelja autorskog prava.

Ništa u ovoj licenci ne narušava ili ograničava autorova moralna prava.

Tekst licence preuzet je s <http://creativecommons.org/>.

Raspored Lige programiranja

- 18.10.2014. – 1. radionica
- 08.11.2014. – **2. radionica**
- 22.11.2014. – **1. kolo Lige programiranja**
- 06.12.2014. – 3. radionica
- 20.12.2014. – 2. kolo Lige programiranja
- termini u 2015. godini će biti naknadno određeni
- Web stranica Lige programiranja:
www.futura.com.hr/liga-programiranja-u-pythonu-2014-2015/

Pravila Lige programiranja

- ☐ Ekipno natjecanje škola **5./6. razredi** i **7./8. razredi**
- ☐ Ekipu škole **5./6. razredi** čine 2 ili 3 učenika iz 5. ili 6. razreda
- ☐ Ekipu škole **7./8. razredi** čine 2 ili 3 učenika iz 7. ili 8. razreda
- ☐ Za rezultate ekipa se uzimaju u obzir bodovi 2 najbolja učenika te ekipe za svako kolo lige
- ☐ Nagrade najboljim ekipama i najboljim pojedincima u kategorijama 5./6. i 7./8. razredi

Ponavljanje gradiva s 1. radionice



□ Posljednji zadatak: izračun prosjeka ocjena

varijabla

```
ocjena1 = int(input('Unesi prvu ocjenu: '))
ocjena2 = int(input('Unesi drugu ocjenu: '))
ocjena3 = int(input('Unesi treću ocjenu: '))
ocjena4 = int(input('Unesi četvrtu ocjenu: '))
prosjek = (ocjena1 + ocjena2 + ocjena3 + ocjena4)/4
print('Prosjek ocjena:', prosjek)
```

print - ispis
teksta i/ili
sadržaja varijabli
na ekran

int - pretvorba
niza znakova u
cijeli broj

input - unos
podataka s
tipkovnice

Relacijski operatori



veće od	>
manje od	<
veće od ili jednako	>=
manje od ili jednako	<=
jednako	==
nije jednako	!=

- Relacijski operatori uspoređuju dva operanda. Rezultat usporedbe ima vrijednosti **True** ili **False** (*Istina* ili *Laž*).

Relacijski operatori



□ Provjeriti kako operatori djeluju:

```
>>> 5 > 4
```

```
True
```

```
>>> 5 < 4
```

```
False
```

```
>>> 5 >= 4
```

```
True
```

```
>>> 5 <= 4
```

```
False
```

```
>>> 5 == 4
```

```
False
```

```
>>> 5 != 4
```

```
True
```

```
>>> a=2
```

```
>>> b=7
```

```
>>> b > a
```

```
True
```

```
>>> b+1 == a*4
```

```
True
```

```
>>> b/a != b//a
```

```
True
```

```
>>> (a+b)**2 <=
```

```
10*a
```

```
False
```

```
>>> r1='tri'
```

```
>>> r2='pet'
```

```
>>> r1 > r2
```

```
True
```

```
>>> m='tri'
```

```
>>> n='Tri'
```

```
>>> m == n
```

```
False
```

```
>>> m > n
```

```
True
```

Nizovi znakova se
uspoređuju slovo po
slovo (uzimajući u obzir
ASCII kod)

Relacijski operatori



□ Redoslijed znakova (ASCII tablica):

	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
p	q	r	s	t	u	v	w	x	y	z	{		}	~	

Zato je:
't' > 'T'

Usput: kako
saznati ASCII
kod nekog slova?

```
>>> ord('T')  
84  
>>> ord('t')  
116
```

Donošenje odluka u programima



- Kako izvesti programsku naredbu (ili više njih) na temelju ispitivanja nekog uvjeta:

```
...  
ako je uvjet onda  
    naredba1_1  
    ...  
    naredba1_n  
...
```

A blue double-headed vertical arrow is positioned to the right of the code block, spanning from the line 'naredba1_1' to the line 'naredba1_n'.

Donošenje odluka u programima



□ U Pythonu:

```
...  
if uvjet:  
    → naredba1_1  
    → ...  
    → naredba1_n  
...
```

Izvršit će se
ako je *uvjet*
zadovoljen
(logički izraz
je istinit)

Svaki redak koji će se izvesti
ako je *uvjet* zadovoljen mora
biti uvučen. Najbolje je
koristiti tipku TAB



□ Primjer:

```
if a>b:  
    print ('A je veće od B')
```

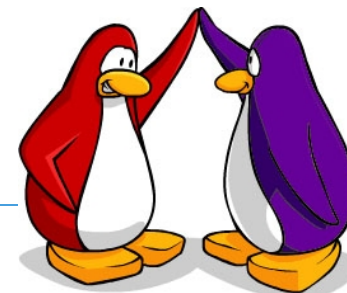
Zadatak: Parni broj



- ☐ Napisati program koji će za uneseni cijeli broj ispisati da li je taj broj paran.
- ☐ Sjećate li se kako se piše i pokreće Python program?



Podsjetnik: Python program



The image shows the Python IDLE environment with three windows and several annotations:

- Python IDLE**: Points to the main application window's menu bar.
- Python IDLE editor**: Points to the code editor window titled '*Python 3.4.1: Untitled*'. It contains the following code:

```
#Jednolinijski komentari
#Ovo je moj prvi Python program.

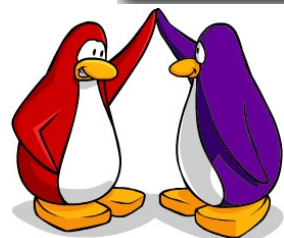
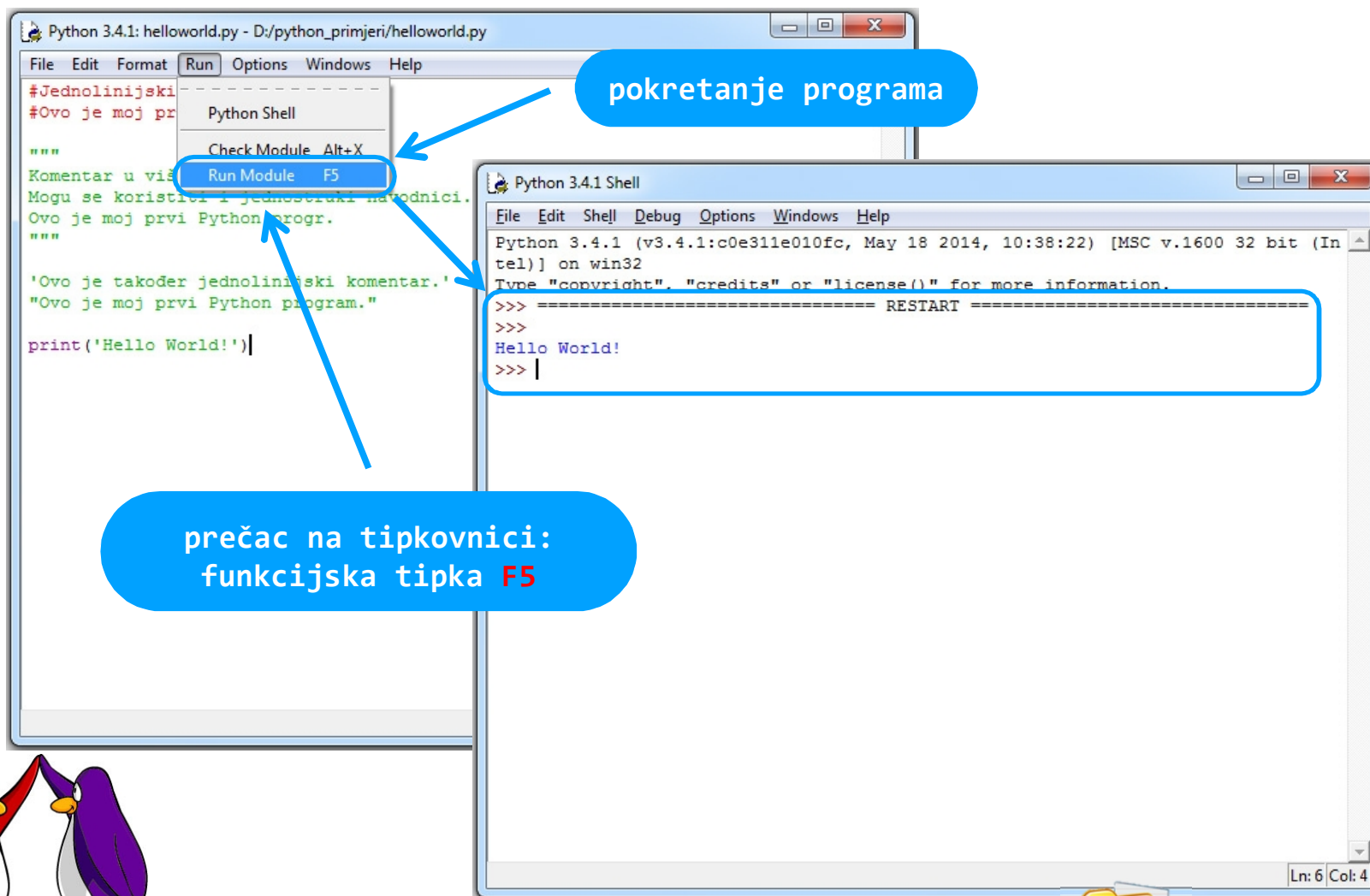
"""
Komentar u više linija.
Mogu se koristiti i jednostruki navodnici.
Ovo je moj prvi Python progr.
"""

'Ovo je također jednolinijski komentar.'
'Ovo je moj prvi Python program.'

print('Hello World!')
```
- komentari**: Points to the multi-line comment block in the code editor.
- ekstenzija .py**: Points to the file extension in the 'Save as type' dropdown of the 'Save As' dialog.
- unos naziva programa**: Points to the 'File name' input field in the 'Save As' dialog, which contains 'helloworld.py'.
- spremanje programa**: Points to the 'Save' button in the 'Save As' dialog.
- kod programa**: Points to the `print('Hello World!')` line in the code editor.

The 'Save As' dialog shows the current directory as 'python_primeri' and the file type as 'Python files (*.py;*.pyw)'.

Pokretanje programa u Python IDLE-u



helloworld.py

Zadatak: Parni broj

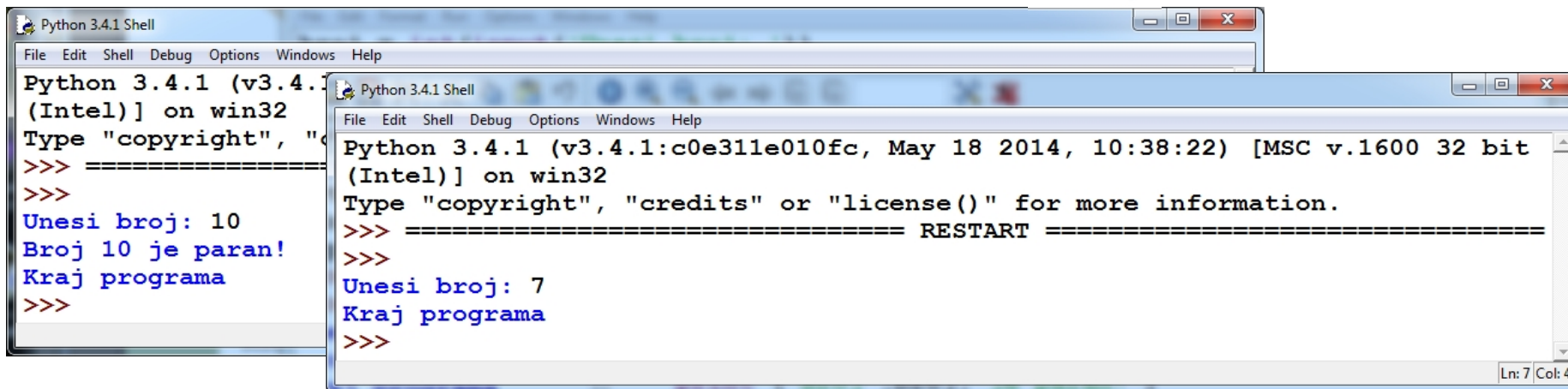


- Napisati program koji će za uneseni cijeli broj ispisati da li je taj broj paran.
- Rješenje:

```
broj = int(input('Unesi broj: '))  
if broj%2 == 0:  
    print ('Broj', broj, 'je paran!')  
print ('Kraj programa')
```



paranBroj.py



```
Python 3.4.1 Shell  
File Edit Shell Debug Options Windows Help  
Python 3.4.1 (v3.4.1:~) [Intel] on win32  
Type "copyright", "credits" or "license()" for more information.  
>>> =====  
>>>  
Unesi broj: 10  
Broj 10 je paran!  
Kraj programa  
>>>  
  
Python 3.4.1 Shell  
File Edit Shell Debug Options Windows Help  
Python 3.4.1 (v3.4.1:c0e311e010fc, May 18 2014, 10:38:22) [MSC v.1600 32 bit  
(Intel)] on win32  
Type "copyright", "credits" or "license()" for more information.  
>>> ===== RESTART =====  
>>>  
Unesi broj: 7  
Kraj programa  
>>>
```

Donošenje odluka u programima



- Često je u programima potrebno odabrati jednu od dvije mogućnosti:

```
...  
ako je uvjet onda  
    naredba1_1  
    ...  
    naredba1_n  
inače  
    naredba2_1  
    ...  
    naredba2_m  
...
```

Izvršit će se
samo JEDAN od
ova dva bloka
naredbi!

Usput: ovakav
način opisa
programa zove se
"pseudojezik" ili
"pseudokod"

Donošenje odluka u programima



□ U Pythonu:

...

if *uvjet*:

naredba1_1

...

naredba1_n

else:

naredba2_1

...

naredba2_m

...

Izvršit će se
samo JEDAN od
ova dva bloka
naredbi!

Donošenje odluka u programima



□ U Pythonu:

```
...  
if uvjet:  
    naredba1_1  
    ...  
    naredba1_n  
  
else:  
    naredba2_1  
    ...  
    naredba2_m  
...
```



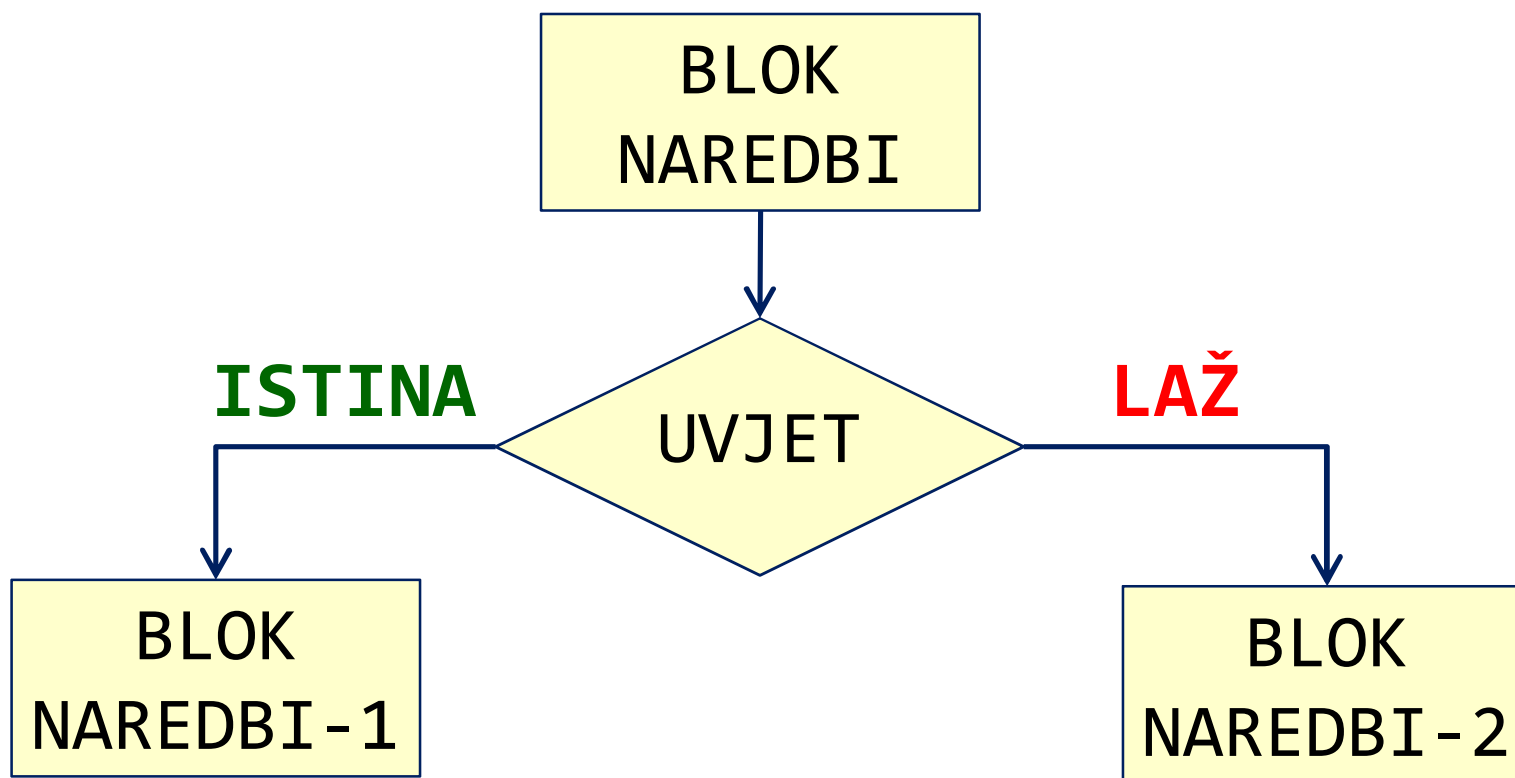
Izvršit će se
ako je *uvjet*
zadovoljen
(logički izraz
je istinit)

Izvršit će se
ako *uvjet* nije
zadovoljen
(logički izraz
je lažan)

Donošenje odluka u programima



- Dakle, na temelju ispitivanja *uvjeta* imamo grananje programa:



Donošenje odluka u programima



- U Pythonu je važno paziti na uvlačenje redaka naredbi koje su dio bloka :

```
...  
if uvjet:  
    → naredba1_1  
    → ...  
    → naredba1_n  
else:  
    → naredba2_1  
    → ...  
    → naredba2_m  
...
```

Izvršit će se
samo JEDAN od
ova dva bloka
naredbi!

Najbolje je
koristiti
tipku TAB

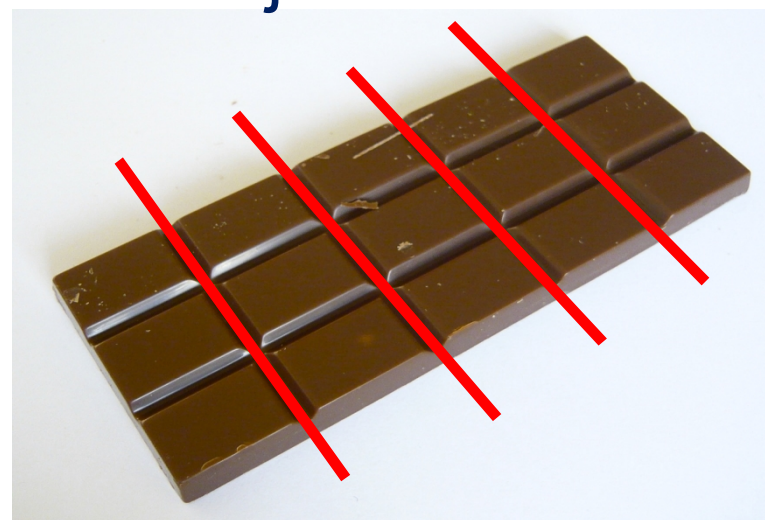


Zadatak: Čokolada



- Antonio želi kupiti čokoladu koja ima toliko kockica da se može ravnomjerno i bez ostatka podijeliti na 5 prijatelja.
- Potrebno je unijeti broj redaka i broj stupaca čokolade, i donijeti odluku da li se sve kockice čokolade mogu podijeliti ravnomjerno.

Primjer: Čokolada
ima 15 kockica –
može se podijeliti
na 5 prijatelja



Zadatak: Čokolada - rješenje



```
brRedova = int(input('Unesi broj redova: '))
brStupaca = int(input('Unesi broj stupaca: '))
if (brRedova*brStupaca)%5 == 0:
    print ('Čokolada se može podijeliti')
    print ('Svatko dobije', (brRedova*brStupaca)//5, 'kocke!')
else:
    print ('Čokolada se ne može ravnopravno podijeliti')
```



cokolada.py

```
Python 3.4.1 Shell
File Edit Shell Debug Options Windows Help
Python 3.4.1 (v3.4.1:c0e311e010fc, May 18 2014, 10:38:22) [MSC
(Intel)] on win32
Type "copyright", "credits" or
>>> =====
>>>
Unesi broj redova: 5
Unesi broj stupaca: 3
Čokolada se može podijeliti
Svatko dobije 3 kocke!
>>>

Python 3.4.1 Shell
File Edit Shell Debug Options Windows Help
Python 3.4.1 (v3.4.1:c0e311e010fc, May 18 2014, 10:38:22) [MSC
(Intel)] on win32
Type "copyright", "credits" or "license()" for more information
>>> ===== RESTART =====
>>>
Unesi broj redova: 4
Unesi broj stupaca: 3
Čokolada se ne može ravnopravno podijeliti
>>> |
```

Donošenje odluka u programima



- U slučaju višestrukog izbora:

```
...  
if uvjet_1:  
    blok naredbi_1  
elif uvjet_2:  
    blok naredbi_2  
...  
elif uvjet_n:  
    blok naredbi_n  
else:  
    blok naredbi  
...
```

Izvršit će se
samo JEDAN od
blokova
naredbi!

Zadatak: tko je viši?



- Dube i Ivo se ne mogu dogovoriti tko je viši, pa to treba riješiti programom. Potrebno je unijeti Dubinu i Ivovu visinu (u cm), i onda ispisati tko je viši i za koliko cm, ili da su ipak jednake visine.



Vrijeme



Zadatak: tko je viši? - rješenje



```
visDube = int(input('Dubina visina: '))
visIvo = int(input('Ivova visina: '))
if visDube > visIvo:
    print ('Dube je viša za', visDube-visIvo, 'cm' )
elif visDube < visIvo:
    print ('Ivo je viši za', visIvo-visDube, 'cm' )
else:
    print ('Ivo i Dube su iste visine!')
```



visina.py

The first screenshot shows the script being run with inputs 155 for Dube and 149 for Ivo, resulting in the output: "Dube je viša za 6 cm".

The second screenshot shows the script being run with inputs 139 for Dube and 151 for Ivo, resulting in the output: "Ivo je viši za 12 cm".

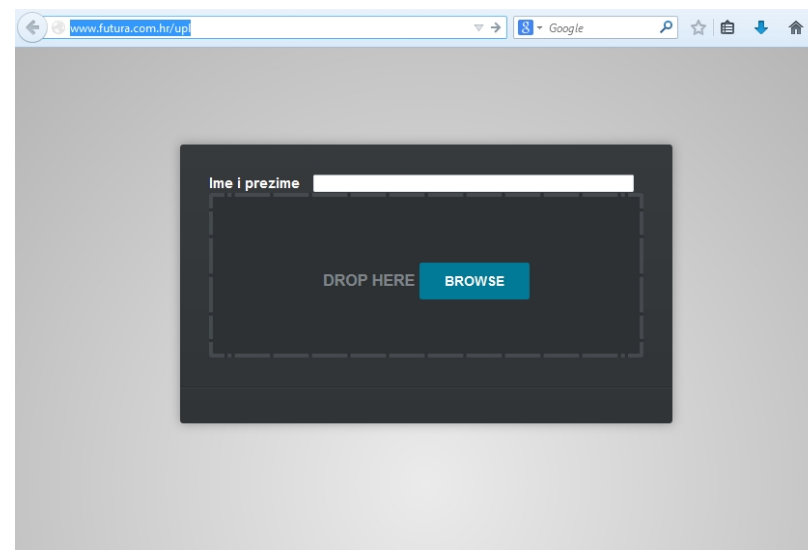
The third screenshot shows the script being run with inputs 162 for both Dube and Ivo, resulting in the output: "Ivo i Dube su iste visine!".

Slanje programa na natjecanju?



Kad idući put bude kolo **Lige programiranja**, bit će potrebno poslati (*upload*) programski kod riješenih zadataka.

Link za slanje programa:
www.futura.com.hr/upl



Slanje programa na natjecanju?



1. Upisati ime

Ime i prezime Dživo Programić

DROP HERE

BROWSE

2. Za svaki program:

- "drag & drop"

ili

- koristiti "browse"



prosjOcjena.py
0.36 KB

Spremanje datoteke prosjOcjena.py [prosjOcjena.py] u arhivu
dzivo programic.zip uspješno !

Logički operatori i logički izrazi



Što ako je uvjet na temelju kojeg treba donijeti odluku složen?

Logički operatori:

logička I operacija	and
logička ILI operacija	or
NE operacija (negacija)	not

Redoslijed izvođenja logičkih operacija:

1.	not
2.	and
3.	or

Logički operatori i logički izrazi



Primjeri logičkih operacija:

```
>>> a=2
>>> b=3
>>> c=10
>>> a>b
False
>>> c>b
True
>>> a>b and c>b
False
>>> a>b or c>b
True
```

Za logičku **AND** operaciju rezultat će biti **True** (istina) samo ako su oba izraza **True** (istinita)

Za logičku **OR** operaciju rezultat će biti **True** (istina) već ako je jedan od izraza **True** (istinit)

Logički operatori i logički izrazi



Kako se podsjetiti ishoda logičkih operacija?

and (i)

```
>>> False and False
False
>>> False and True
False
>>> True and False
False
>>> True and True
True
```

or (ILI)

```
>>> False or False
False
>>> False or True
True
>>> True or False
True
>>> True or True
True
```

not (ne)

```
>>> not False
True
>>> not True
False
```

- ☐ Je li isto **True** i **true**? Ili **False** i **false**?
- ☐ Što je rezultat izraza: **True and true**

Logički operatori i logički izrazi



Primjeri logičkih operacija:

```
>>> a=2
>>> b=3
>>> c=10
>>> a+2*3>=c or not(a>b) and a*b-2==c%6
```

1.	aritmetički
2.	relacijski
3.	logički

Ako imamo kombinirane aritmetičke, relacijske i logičke operatore, onda je ovo redoslijed izvođenja operacija.

Logički operatori i logički izrazi



Primjeri logičkih operacija:

```
>>> a=2
>>> b=3
>>> c=10
>>> a+2*3>=c or not(a>b) and a*b-2==c%6
True

>>> (a+2*3>=c) or (not(a>b) and (a*b-2==c%6))
True
```

Ako ipak nismo posve sigurni
u redosljed operacija onda
je najbolje koristiti
zagrade!

Zadatak: Zaključivanje ocjena



- Napisati program koji će na temelju unešene prosječne ocjene (na dva decimalna mjesta) ispisati zaključenu ocjenu.

dovoljan $\Rightarrow$ prosjek ≥ 2.00 i prosjek < 2.50

dobar $\Rightarrow$ prosjek ≥ 2.50 i prosjek < 3.50

vrlo dobar $\Rightarrow$ prosjek ≥ 3.50 i prosjek < 4.50

odličan $\Rightarrow$ prosjek ≥ 4.50 i prosjek ≤ 5.00

Ako prosjek nije među navedenim vrijednostima javiti grešku!

	I.	II.	III.	IV.	V.	VI.
1		2		1		
2	1	1	1	1	2	
3		2	1			

Vrijeme



Rješenje: Zaključivanje ocjena



```
prosjoelj = float(input('Prosječna ocjena: '))
if prosjoelj >= 2.00 and prosjoelj<2.50:
    print ('Dovoljan')
elif prosjoelj >= 2.50 and prosjoelj<3.50:
    print ('Dobar')
elif prosjoelj >= 3.50 and prosjoelj<4.50:
    print ('Vrlo dobar')
elif prosjoelj >= 4.50 and prosjoelj<=5.00:
    print ('Odličan')
else:
    print ('Pogrešna prosječna ocjena!')
```



prosjoeljena.py

Testirati program s dovoljno vrijednosti – npr.:

1.5 2.25 2.5 3 3.5 3.99 4.5 5 17

Ne zaboravite!

- ☐ Za 15 dana – **u subotu 22.11.2014.** –
1. kolo Lige programiranja
- ☐ **5./6. razredi** ekipe od 3 učenika (ili barem 2)
početak **9:00**
- ☐ **7./8. razredi** ekipe od 3 učenika (ili barem 2)
početak **10:30**
- ☐ **3** zadatka rješavate **75** minuta
- ☐ nemojte kasniti!

