

Sveučilište u Rijeci  
Odjel za informatiku  
Akademska 2016./2017. godina  
Mentor: dr.sc. Nataša Hoić-Božić, dr.sc. Martina Holenko Dlab

## Priprema za izvođenje nastavnog sata informatike

### **Python: Naredba odlučivanja *for***

Maja Manojlović

Rijeka, 28. studenog 2016.

**Škola:** Prva riječka hrvatska gimnazija

**Razredni odjel:** 2.a

**Nastavni predmet:** Informatika

**Nastavna cjelina:** Programiranje u Pythonu

**Nastavna tema:** Naredbe odlučivanja

**Nastavna jedinica:** Naredba odlučivanja *for*

**Tip nastavnog sata:** Kombinirani sat obrade novog gradiva i vježbanja

**Vrijeme izvođenja:** 4. sat

**Mjesto izvođenja:** kabinet s računalima 006

**Cilj (svrha) sata:** Upoznati učenike sa naredbom za ponavljanje *for* u Pythonu te pokazati primjenu na zadacima.

**Ishodi učenja:**

Učenici će moći/bit će sposobni:

- Uočiti naredbu za ponavljanje *for* u zadanom programskom kôdu
- Napisati sintaksu naredbe za ponavljanje *for*
- Objasniti korake izvođenja programa s naredbom ponavljanja *for*
- Zaključiti kada primijeniti naredbu ponavljanja *for*
- Samostalno rješavati zadatke koristeći naredbu za ponavljanje *for*

**Nastavna načela:**

- **načelo postupnosti** (novo gradivo izlaže se postupno, teorija se potkrepljuje primjerima i zadacima počevši od jednostavnijih ka složenima, novi sadržaji povezuju se sa prethodno naučenim sadržajima, na kraju sata ponavljaju se najvažniji koncepti)
- **načelo primjerenosti** (primjeri i zadaci primjereni su dobi učenika i njihovom predznanju)
- **načelo zornosti** (grafičkim prikazom predočava se tijek naredbe *for*, animacijom se vizualno ističu najbitniji dijelovi sintakse naredbe *for*)
- **načelo individualizacije** (za vrijeme praktičnog rada učenici samostalno rješavaju zadatke dok profesorica nadgleda proces te pruža pomoć pri rješavanju zadataka ukoliko je ona potrebna)

**Nastavne metode:** metoda razgovora, metoda usmenog izlaganja, metoda demonstracije, metoda praktičnih radova, metoda pisanja i rada s tekstom

## ORGANIZACIJA NASTAVNOG SATA

ETAPA	SADRŽAJ	OBLICI RADA	METODE RADA	VRIJEME
Uvodni dio	<ul style="list-style-type: none"><li>– priprema za rad</li><li>– najava teme sata</li><li>– motiviranje učenika</li></ul>	frontalni	usmeno izlaganje	3 min
Središnji dio	<ul style="list-style-type: none"><li>– obrada gradiva</li><li>– demonstracija primjera</li><li>– rješavanje zadataka</li></ul>	frontalni, frontalni s elementima individualizacije, individualni	usmeno izlaganje, demonstracija, razgovor, praktični rad na računalu, metoda pisanja i rada na tekstu	32 min
Završni dio	<ul style="list-style-type: none"><li>– ponavljanje</li><li>– zadavanje domaće zadaće</li></ul>	frontalni	usmeno izlaganje, razgovor	5 min

## TIJEK NASTAVNOG SATA

### Uvodni dio

Nakon pripreme za rad, ukratko se prisjeća gradiva sa prethodnih sati te motivira učenike za daljnji rad.

Do sada su učenici već naučili da se naredbe mogu izvršavati slijedno jedna iza druge, ali i da se naredbe mogu izvršavati samo ukoliko su ispunjeni određeni uvjeti. Neke od naredbi odlučivanja su već poznate učenicima, poput naredbe *if*. Pomoću njih možemo napisati programe koji se ponašaju „razumno“, odnosno donose sami odluke. Na današnjem satu obrađivati će se još jedna naredba odlučivanja, a to je naredba *for*. Učenicima se postavljaju pitanja: „Što se događa kada ne želimo da program sam donosi odluke o izvođenju neke naredbe? Možemo li onda koristiti, primjerice, naredbu *if* ili neke od ostalih naredbi koje smo do sada obrađivali?“ i „Da li ste mogli upisati srednju školu prije nego li ste završili osam razreda osnovne škole?“ Ponekad želimo da program izvrši neku naredbu ili blok naredbi točan broj puta i da se nakon toga nastavi izvoditi program prvom naredbom iza bloka. Naredba pomoću koje se to može izvršiti je naredba *for*. Zapisuje se naslov na ploču: „Naredba odlučivanja *for*“.

## Središnji dio

Prije uvođenja opće sintakse naredbe *for* u Pythonu, uz pomoć prezentacije se prikazuje njen općeniti zapis u pseudojeziku:

```
PSEUDOJEZIK

za i := 1 do n činiti
{
    naredba 1;
    naredba 2;
    ...
    naredba n;
}
```

Potrebno je uočiti da će se naredba izvršiti od 1 do n puta, odnosno da je broj ponavljanja naredbe strogo određen, a nakon toga će se program nastaviti izvoditi prvom naredbom iza bloka. Zatim se prikazuje sintaksa naredbe *for* u programskom jeziku Python koju učenici moraju zapisati u svoje bilježnice:

```
PYTHON

for i in range (n):
    naredba 1
    naredba 2
    ...
    naredba n
```

Potiče se učenike da primijete sličnosti i razlike između dva zapisa. Podsjetimo se da se uvjet završava znakom dvotočke, a blok naredbi se ne ograđuje zagradama kao što je to slučaj u pseudojeziku. Također se naglašava kako je bitno da blok naredbi nakon uvjeta bude uvučen. Objašnjava se da  $i$  označava nepoznanicu, odnosno brojač petlje dok  $n$  označava broj ponavljanja brojača. Ističu se ključni dijelovi sintakse te naglašava da Python počinje brojiti s nulom, ukoliko nije drugačije određeno. Iz tog razloga najveći broj koji će dosegnuti varijabla  $i$  nije  $n$  već  $n-1$ .

Kako bi učenici lakše razumjeli strukturu i primjenu petlje *for*, na ploči se rješava jednostavan primjer u suradnji s učenicima.

**Primjer 1:**

Napišite program kojim ćete koristeći *for* petlju ispisati vrijednost brojača do broja 5.

**Rješenje:**

```
for i in range (6):  
    print (i)
```

**Način rješavanja:**

Od učenika se traži da navedu općeniti oblik sintakse *for* petlje. Zapisuje se na ploču:

```
for i in range ():  
    print (i)
```

Prostor za upis uvjeta ostavlja se prazan te se od učenika očekuje odgovor na pitanje: „Koliko puta moramo ispisati brojač  $i$ ?“

Kada učenici daju točan odgovor (broj 6), on se zapisuje na predviđeno mjesto u sintaksi. Potrebno je još jednom naglasiti, posebno ako neki učenici daju odgovor 5, da se brojač u Pythonu kreće od nule te da će njegova krajnja vrijednost biti  $n-1$  (u ovom slučaju 6-1). Kada bi napisali:

```
for i in range (5):  
    print (i)
```

program bi ispisao brojač samo 4 puta. Zaključujemo da je funkcija *print (i)* pozvana šest puta i pritom ispisala šest vrijednosti varijable  $i$ , počevši od nule. To znači da će u zadnjem prolazu kroz petlju vrijednost brojača biti za 1 manja, a petlja će se izvršavati sve dok brojač ne poprimi vrijednost 6. Prilikom izvršavanja petlje, u svakom će se prolazu ispisati vrijednost brojača, osim u zadnjem jer se petlja zaustavlja.

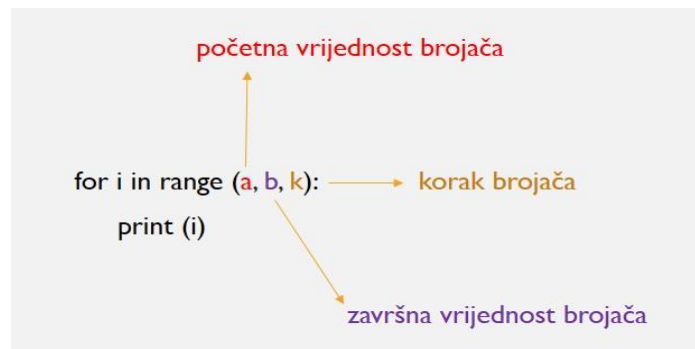
Učenici u bilježnice moraju prepisati rješenje primjera. Zatim profesorica otvara Python IDLE u kojem pokazuje unaprijed pripremljen kod za primjer 1 i pokreće program naredbom *Run* | *Run Module* ili F5. Pokazuje se rezultat izvršavanja programa.

Ovaj način pisanja koda ispisati će nam rješenje u jednom stupcu, sa vrijednostima jednom ispod druge. Ako želimo da nam se rješenje ispisuje u retku kod će izgledati ovako:

```
for i in range (6):  
    print (i, end = ' ')
```

Na prezentaciji se prikazuje dijagram tijeka navedenog primjera zajedno sa objašnjenjima svakog koraka.

Ako u *for* petlji pozovemo funkciju *range()* u obliku *range (a, b, k)*, kontrolna varijabla petlje poprimati će vrijednosti:  $a$ ,  $a+k$ ,  $a+2\cdot k$ , ... pri čemu će zadnja vrijednost kontrolne varijable biti strogo manja od  $b$ . Ovakav oblik petlje također prikazujemo sintaksom:



Bojama se ističu novi elementi sintakse radi lakšeg uočavanja i razlikovanja. Kada se koristi ovaj oblik *for* petlje, naredba se ponavlja sve dok se početna vrijednost petlje ( $a$ ), ne izjednači sa krajnjom vrijednosti petlje ( $b$ ). Ističe se kako pravilo krajnje vrijednosti  $n-1$ , u ovom slučaju  $b-1$ , i dalje vrijedi. Korak petlje može se, ali i ne mora navesti. Ukoliko ne navedemo vrijednost koraka petlje, ona se automatski postavlja na vrijednost 1 i tada se vrijednost brojača u svakom prolazu petlje povećava za 1. Ako želimo da se vrijednost brojača povećava za proizvoljan broj puta, upisati ćemo vrijednost koraka.

Zajednički se rješava još jedan primjer kako bi učenici upamtili na koji način se koristi funkcija *range (a, b, k)* te koja je uloga koraka petlje.

### Primjer 2:

Pomažete prijatelju u selidbi. U autu se nalazi 20 spakiranih kutija. Morate donijeti iz auta u stan svaku treću kutiju. Napišite program koji će ispisati broj kutije jedan ispod drugog.

### Rješenje:

```
for i in range (1,21,3):  
    print ('Unosimo kutiju broj', i)
```

## Način rješavanja

Učenike se potiče da promotre zadatak. Možemo primijetiti da nije potrebno ispisati broj svake kutije već svake druge. Učenicima se naglašava da je u ovom primjeru potrebno koristiti funkciju *range* u obliku *range(a, b, k)* na način da im se postavlja pitanje poput: „Što je prvo što možemo zaključiti iz zadatka? Koji oblik *for* petlje trebamo koristiti?“, a zatim se pitamo: „Da li je potrebno odrediti korak petlje?“ Potiče se učenike na aktivno sudjelovanje te se njihovi odgovori zapisuju na ploču. Ističe se kako u ovom primjeru nije korištena naredba za ispis u jednom retku.

Učenici u bilježnice moraju prepisati rješenje primjera. Zatim profesorica otvara Python IDLE u kojem pokazuje unaprijed pripremljen kod za primjer 2 i pokreće program naredbom *Run* | *Run Module* ili F5. Pokazuje se rezultat izvršavanja programa.

U nastavku sata učenici će samostalno rješavati zadatke za vježbu. Profesorica obilazi učenike i provjerava da li svi rješavaju zadatke te pruža pomoć u rješavanju ukoliko je ona potrebna. Učenike se podsjeća da, nakon rješavanja pojedinog zadatka, rješenje (datoteku pisanu u programu Python IDLE) pohrane na USB memoriju kako bi se mogli koristiti njime i kod kuće. Svaki od zadataka riješiti će jedan učenik na glavnom računalu.

### **Zadatak 1:**

Napišite program koji od korisnika traži da unese prirodan broj. Koristeći *for* petlju ispišite svaki drugi broj od 0 do broja koji je unio korisnik.

### **Rješenje:**

```
n = int(input('Unesite prirodan broj: '))
for i in range(0, n+1, 2):
    print(i)
```

## Način rješavanja:

Najprije se od korisnika traži da unese prirodan broj koji se zatim pridružuje varijabli proizvoljnog imena:

```
n = int(input('Unesite prirodan broj: '))
```

Primijetimo da ovaj zadatak traži korištenje funkcije *range(a, b, k)* što znači da je potrebno definirati početnu i završnu vrijednost brojača, kao i korak brojača. Početna vrijednost je zadana u zadatku i iznosi 0, a na mjesto završne vrijednosti upisuje se varijabla koju su učenici proizvoljno odredili. Korak petlje biti će 2 zato što se u zadatku traži ispis svakog trećeg broja. Primjećujemo da program mora izvršavati naredbu do broja koji je unio korisnik. Podsjetimo se da će najveća vrijednost koju će dostići varijabla *i* biti *n-1*. Iz tog razloga završnu vrijednost brojača zapisujemo kao *n+1*.

```
for i in range (0, n+1, 2):
```

Konačno rješenje glasilo bi:

```
n = int(input('Unesite prirodan broj: '))
for i in range (0, n+1, 2):
    print (i)
```

Kada većina učenika riješi zadatak, nastavnica proziva jednog od učenika da riješi zadatak na glavnom računalu. Tijekom rješavanja učenik objašnjava korake zadatka, a nastavnica mu pomaže ukoliko je to potrebno. Pokreće se program i prikazuje ispis koji dobijemo rješavanjem zadatka.

Zatim se rješava sljedeći zadatak.

**Zadatak 2:**

Napišite program za ispis dvostruke vrijednosti prvih  $n$  prirodnih brojeva.

**Rješenje:**

```
n = int(input('Unesite prirodan broj: '))
for i in range (1, n+1):
    print ('Dvostruka vrijednost od', i, 'je', 2*i)
```

**Način rješavanja:**

U ovom zadatku također je potrebno zatražiti od korisnika upis prirodnog broja:

```
n = int(input('Unesite prirodan broj: '))
```

Primijetimo da u ovom zadatku nije potrebno određivati korak petlje, ali je bitno naglasiti da se završna vrijednost brojača također zapisuje u obliku  $n+1$ :

```
for i in range (1, n+1):
```

Vidimo da program treba ispisati dvostruku vrijednost prvih  $n$  prirodnih brojeva. To znači da svaku vrijednost brojača  $i$  moramo pomnožiti sa brojem 2. Ovaj uvjet postavlja se u naredbi *print*.

```
print ('Dvostruka vrijednost od', i, 'je', 2*i)
```

I ovoga puta jedan učenik dolazi na glavno računalo riješiti zadatak. Tijekom rješavanja učenik objašnjava korake zadatka, a nastavnica mu pomaže ukoliko je to potrebno. Pokreće se program i prikazuje ispis koji dobijemo rješavanjem zadatka.



Nakon toga, rješava se treći zadatak.

**Zadatak 3:**

Napišite program koji zbraja prvih 20 parnih brojeva. Ispis neka bude vodoravan, a brojevi odvojeni točkom.

**Rješenje:**

```
zbroj=0
for i in range (0,21,2):
    zbroj=zbroj+i
    print ('Nakon broja', i, 'zbroj je', zbroj, end=' .')
```

**Način rješavanja:**

U trećem zadatku važno je uočiti da je potrebno koristiti matematičku operaciju zbrajanja. Iz tog razloga potrebno je na početku programa uvesti novu varijablu *zbroj* u koju će se „spremati“ zbroj prvih 20 parnih brojeva. Početna vrijednost varijable *zbroj* postavljena je na 0.

Početna vrijednost brojača je 0, a završna 21. Također je potrebno navesti i korak brojača kako bi nam program zbrajao svaki drugi broj, odnosno samo parne brojeve. Korak brojača stoga postavljamo na 2.

```
for i in range (0,21,2):
```

Poslije svake nove vrijednosti brojača *zbroj* se treba povećati za vrijednost brojača. To se prikazuje pomoću matematičkog operatora zbrajanja:

```
zbroj=zbroj+i
```

U posljednjem dijelu zadatka potrebno je vodoravno ispisati *zbroj* nakon svake vrijednosti brojača, a ispise odvojiti točkom.

```
print ('Nakon broja', i, 'zbroj je', zbroj, end=' .')
```

Također je važno naglasiti da je naredba `print` postavljena unutar petlje *for* kako bi se mogli ispisati svi međuzbrojevi. U protivnom bi se ispisao samo *zbroj* nakon posljednjeg broja kroz koji prolazi brojač, a to je broj 20.

Ukoliko ostane vremena, učenici će riješiti i jedan dodatni zadatak.

**Zadatak 4 (dodatni):**

Napišite program koji će prebrojati koliko je brojeva djeljivih sa 13 u intervalu od 5 do 150.

**Rješenje:**

```
brojac=0
for i in range (5,150):
    if i%13==0:
        brojac=brojac+1
print ('Brojeva djeljivih s 13 ima', brojac)
```

Učenicima se naglašava kako je i u ovom zadatku potrebno uvesti novu varijablu *brojac* u koju će se „spremati“ podatci o brojevima koji su djeljivi sa 13. Također, treba napomenuti kako će se u ovom zadatku koristiti i naredba *if*. Pomoću nje program će odlučiti da li je broj djeljiv sa 13 ili ne. Potom će se nakon svakog prolaza kroz naredbu varijabla *brojac* uvećati za 1, ukoliko je broj djeljiv sa 13. Naredba *print* zapisuje se unutar *for* petlje, ali izvan naredbe *if*.

**Završni dio**

U završnom dijelu sata organizirano je kratko ponavljanje uz pomoć prezentacije kako bi učenici još jednom ponovili najbitnije koncepte sa današnjeg sata. Učenici će odgovarati na pitanja koja će se prikazivati na prezentaciji. Po potrebi, prozivati će se učenike da dođu na ploču odgovoriti na pojedine zadatke.

**PONAVLJANJE**

- **Kako glasi općeniti zapis naredbe odlučivanja *for*?**

```
for i in range (n):
    print (i)
```

- **Kako glasi njen „prošireni” zapis?**

```
for i in range (a, b, k):
    print (i)
```

## PONAVLJANJE

- Što u funkciji **range (a, b, k)** označava **a**, što **b**, a što **k**?

**a** = početna vrijednost brojača

**b** = završna vrijednost brojača

**k** = korak brojača

## PONAVLJANJE

- Što će se ispisati nakon izvođenja sljedećeg programa?

```
s=0
```

```
for i in range (5):
```

```
    s=s+2*i
```

```
print (s)
```

➤ Ispisati će se broj 20.

## PONAVLJANJE

- Što će se ispisati nakon izvođenja sljedećeg programa?

```
for i in range (1,11):
```

```
    if i !=7:
```

```
        print (i)
```

➤ Ispisati će se svi brojevi iz intervala od 1 do 10 koji su različiti od 7.

Nakon ponavljanja učenicima se podijele nastavni listići sa domaćom zadaćom koju sljedeći put moraju donijeti na USB memoriji na nastavu kako bi zajedno provjerili rješenja. Zadaci se zajedno pročitaju i prokomentiraju te, ako je potrebno, pojašne.

Na kraju nastavnog sata, nastavnica najavljuje gradivo koje će se obrađivati sljedeći sat. Upoznati ćemo se sa još jednom naredbom odlučivanja, naredbom *while*, a nakon toga prijeći ćemo na ugrađene funkcije i formule za stringove koje se koriste u Pythonu.

## DOMAĆA ZADAĆA

### 1. Zadatak

Napišite program koji izračunava umnožak prvih 10 prirodnih brojeva.

#### Rješenje:

```
produkt = 0
for i in range(1, 11):
    produkt = produkt*i
    print('Produkt je:', produkt)
```

### 2. Zadatak

Što će ispisati sljedeći dio programa? Objasni zašto.

```
s=1
for i in range(1, 10,2):
    s*=i
print(s)
```

#### Rješenje:

Ispisati će broj 945. Program prolazi intervalom od 1 do 10 i pritom uzima svaku drugu vrijednost i množi ju sa vrijednosti varijable *s*. U svakom prolazu umnožak se sprema u varijablu *s*. Na kraju se ispisuje konačni umnožak (zato što funkcija *print* nije uvučena, već je izvan *for* petlje) nakon što brojač dođe do kraja petlje, odnosno do broja 10.

### 3. Zadatak

Modificiraj kod iz prethodnog zadatka tako da brojač prođe kroz svaku vrijednost u intervalu od 1 do 10 i da se vrijednost brojača nakon svakog prolaza ispiše na ekranu.

#### Rješenje:

```
s=1
for i in range(1, 10):
    s*=i
    print(s)
```

### 4. Zadatak

Napišite program koji ispisuje prirodne brojeve od 3 do 28 unazad.

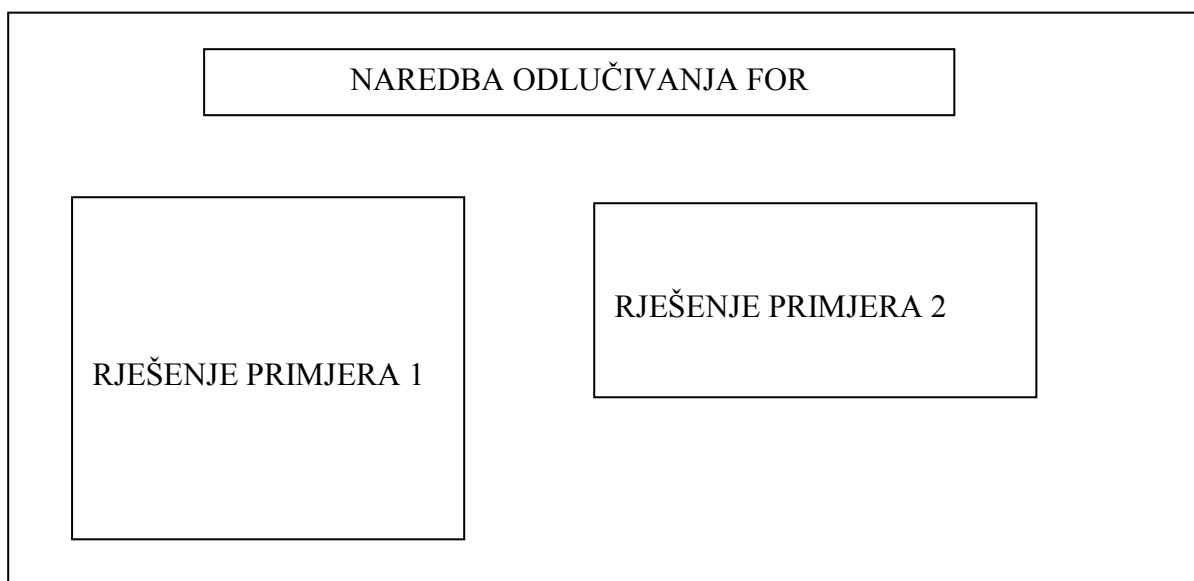
#### Rješenje:

```
for i in range(28, 2, -1):
    print(i)
```

## NASTAVNA SREDSTVA I POMAGALA

- literatura za pripremanje nastavnika:
  - Budin, L. i ostali.: Rješavanje problema programiranjem u Pythonu, Element, Zagreb, 2012.
  - <http://os-zakanje.skole.hr/upload/os-zakanje/images/multistatic/52/File/Python/Python6-3.pdf>, (25.11.2016.)
  - <https://sites.google.com/site/sandasutalo/osnove-programiranja/programska-petlja/petlja-for>, (25.11.2016.)
  - [http://www.znanje.org/knjige/computer/python/081python/081\\_for\\_alg\\_pyth\\_1.php?sifra=08111002.htm](http://www.znanje.org/knjige/computer/python/081python/081_for_alg_pyth_1.php?sifra=08111002.htm) (25.11.2016.)
- obvezna literatura učenika:
  - Budin, L. i ostali.: Rješavanje problema programiranjem u Pythonu, Element, Zagreb, 2012.
- PowerPoint prezentacija (koristi nastavnik u svrhu prezentacije gradiva i motivacije)
- program Python IDLE
- računalo s projektorom (koristi nastavnik za izvođenje prezentacije)
- flomaster i ploča
- računala (koriste učenici za rješavanje zadataka u Pythonu)

## PLANIRANJE RADA NA ŠKOLSKOJ PLOČI



## Prilog 1

### Nastavni listić

#### DOMAĆA ZADAĆA

##### 1. Zadatak

Napišite program koji izračunava umnožak prvih 10 prirodnih brojeva.

##### 2. Zadatak

Što će ispisati sljedeći dio programa? Objasni zašto.

```
s=1
```

```
for i in range (1, 10,2):
```

```
    s*=i
```

```
print(s)
```

##### 3. Zadatak

Modificiraj kod iz prethodnog zadatka tako da brojač prođe kroz svaku vrijednost u intervalu od 1 do 10 i da se vrijednost brojača nakon svakog prolaza ispiše na ekranu.

##### 4. Zadatak

Napišite program koji ispisuje prirodne brojeve od 3 do 28 unazad.

<b>Datum:</b>	
<b>Predavanje slušali studenti:</b>	

<b>Vrednovanje predavanja od strane mentora:</b>	<b>Element</b>	<b>Bodovi</b>
	Obrada teme i stručnost	/5
	Primjena odgovarajućih nastavnih metoda	/5
	Struktura i trajanje nastavnog sata	/4
	Primjeri i zadaci	/4
	Način izlaganja	/3
	Interakcija s učenicima	/3
	PowerPoint prezentacija	/3
	Pisana priprema	/3
	<b>Ukupno:</b>	<b>/30</b>
<b>Komentari:</b>		
<b>Potpis mentora:</b>		

# NAREDBA ODLUČIVANJA FOR

Maja Manojlović

Naredba odlučivanja for

27/11/16

1

## SINTAKSA PETLJE FOR

### PSEUDOJEZIK

```
za i := 1 do n činiti
{
    naredba 1;
    naredba 2;
    ...
    naredba n;
}
```

Naredba odlučivanja for

### PYTHON

```
for i in range(n):
    naredba 1
    naredba 2
    ...
    naredba n
```

27/11/16

2

## PRIMJER 1 (1/2)

Napišite program kojim ćete koristeći *for* petlju ispisati vrijednost brojača do broja 5.

### > RJEŠENJE:

```
for i in range(6):
    print(i)
```

```
for i in range(6):
    print(i, end = ' ')
```

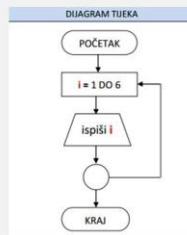
Naredba odlučivanja for

27/11/16

3



## PRIMJER I (2/2)

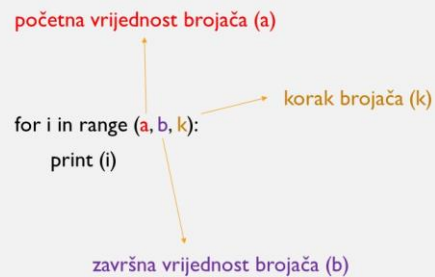


- određujemo početnu i završnu vrijednost brojača
- ispisuje se svaka trenutna vrijednost brojača
- radnja se ponavlja dok se ne dođe do krajnje vrijednosti brojača

Naredba odlučivanja for

27/11/16

4



Naredba odlučivanja for

27/11/16

5

## PRIMJER 2:

Pomažete prijatelju u selidbi. U autu se nalazi 20 spakiranih kutija. Morate donijeti iz auta u stan svaku treću kutiju. Napišite program koji će ispisati broj kutije jedan ispod drugog.

### ➤ RJEŠENJE:

```
for i in range (1,21,3):  
    print ('Unosimo kutiju broj', i)
```

Naredba odlučivanja for

27/11/16

6

### ZADATAK 1:

Napišite program koji od korisnika traži da unese prirodan broj. Koristeći *for* petlju ispišite svaki treći broj od 0 do broja koji je unio korisnik.

#### ➤ RJEŠENJE:

```
n = int(input('Unesite prirodan broj: '))
for i in range(0, n+1, 3):
    print(i)
```

Naredba odlučivanja for

27/11/16

7

### ZADATAK 2:

Napišite program za ispis dvostruke vrijednosti prvih n prirodnih brojeva.

#### ➤ RJEŠENJE:

```
n = int(input('Unesite prirodan broj: '))
for i in range(1, n+1):
    print('Dvostruka vrijednost od', i, 'je', 2*i)
```

Naredba odlučivanja for

27/11/16

8

### ZADATAK 3:

Napišite program koji zbraja prvih 20 parnih brojeva. Ispis neka bude vodoravan, a brojevi odvojeni točkom.

#### ➤ RJEŠENJE:

```
zbroj=0
for i in range(0,21,2):
    zbroj=zbroj+i
print('Nakon broja', i, 'zbroj je', zbroj, end=' .')
```

Naredba odlučivanja for

28/11/16

9

## ZADATAK 4 (DODATNI)

Napišite program koji će prebrojati koliko je brojeva djeljivih sa 13 u intervalu od 5 do 150.

### ➤ RJEŠENJE:

```
brojac=0
for i in range (5,150):
    if i%13==0:
        brojac=brojac+1
print ('Brojeva djeljivih s 13 ima', brojac)
```

Naredba odlučivanja for

28/11/16

10

## PONAVLJANJE (1/4)

### • Kako glasi općeniti zapis naredbe odlučivanja for?

```
for i in range (n):
    print (i)
```

### • Kako glasi njen „prošireni” zapis?

```
for i in range (a, b, k):
    print (i)
```

Naredba odlučivanja for

28/11/16

11

## PONAVLJANJE (2/4)

### • Što u funkciji **range (a, b, k)** označava **a**, što **b**, a što **k**?

**a** = početna vrijednost brojača

**b** = završna vrijednost brojača

**k** = korak brojača

Naredba odlučivanja for

28/11/16

12

### PONAVLJANJE (3/4)

• Što će se ispisati nakon izvođenja sljedećeg programa?

```
s=0
for i in range(5):
    s=s+2*i
print(s)
```

➤ Ispisati će se broj 20.

Naredba odlučivanja for

28/11/16

13

### PONAVLJANJE (4/4)

• Što će se ispisati nakon izvođenja sljedećeg programa?

```
for i in range(1,11):
    if i !=7:
        print(i)
```

➤ Ispisati će se svi brojevi iz intervala od 1 do 10 koji su različiti od 7.

Naredba odlučivanja for

28/11/16

14