

# Informatika i programiranje

Doc. dr. sc. Jurica Babić

# Sadržaj

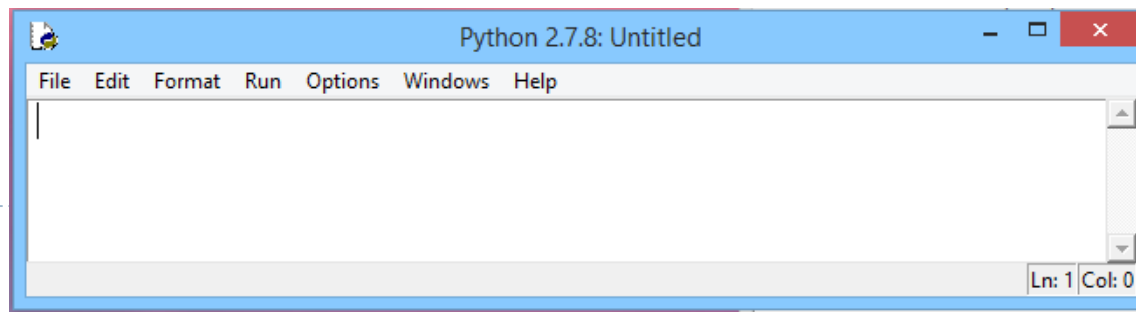
---

- ▶ Program
- ▶ Interakcija s korisnikom (input())
- ▶ Petlje u kornjačinoj grafici
- ▶ Petlje
- ▶ Račvanje

# Uređivačko sučelje - editor

---

- ▶ Za trajno čuvanje naredbi i programa koje smo osmislili
- ▶ Pojednostavljuje ispravljanje pogrešaka
  - File → New File
  - Ctrl + N
- ▶ Novi prozor s naslovom **Untitled**
- ▶ Pohranjivanje u datoteku: File → Save as...
- ▶ Upisati ime (naziv programa) i sufiks **.py** → ova datoteka zove se i **MODUL**
- ▶ Upisani naziv će se pojaviti u naslovnoj traci programa



# Otvaranje spremljenog modula

---

- ▶ Iz Phyton Shell-a: **File** → **Open**
- ▶ **CTRL + O**
- ▶ Desnim klikom na ikonu programa i izborom **Edit with IDLE**

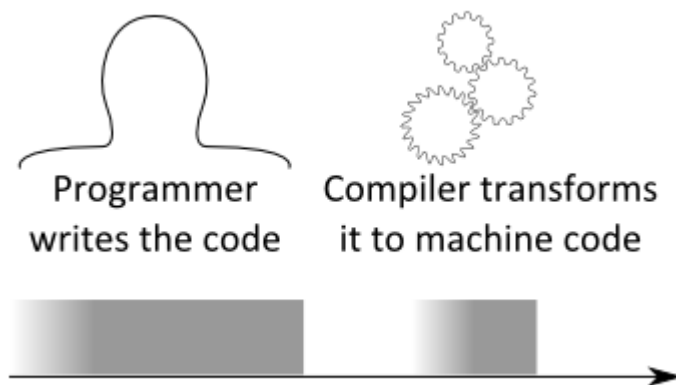
# Pokretanje spremljenog programa

---

- ▶ Run → Run module
- ▶ F5
  
- ▶ Phyton Sell je postao ulazno-izlazni prozor = interaktivno sučelje:
  - ▶ Ako vaš program čeka korisnika da unese neki podatak – taj podatak će korisnik upisati u Phyton Shell prozoru
  - ▶ Ako vaš program mora ispisati neki rezultat – taj podatak će se ispisati u Phyton Shell prozoru

# Program

- ▶ **Računalni program** je skup uputa računalu što treba učiniti i kako to izvesti.
- ▶ Računalni program se pomoću jezičnog prevoditelja (engl. compiler) prevodi u binarni brojevni sustav kako bi ga računalo razumjelo i izvršilo.
- ▶ Jezični prevoditelj (kompajler) je program koji čita program napisan na izvornom jeziku (Python), te ga prevodi na strojni jezik.

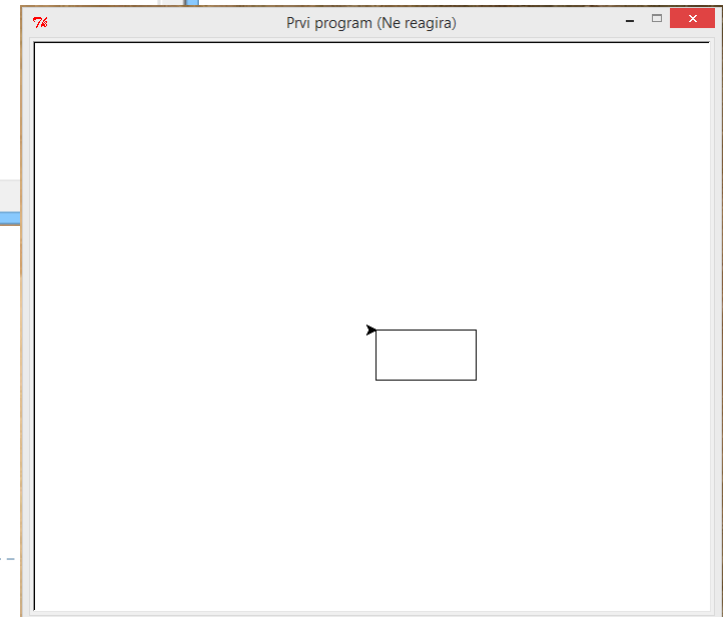


# Primjer: pravokutnik

- ▶ Napišite program koji će nacrtati pravokutnik stranica duljine 100 i 50.

```
Python 2.7.8: 2pred_pravokutnik.py - C:/Razno/Posao/Nastava/Informatika i programiranje/2pred_... - □ ×
File Edit Format Run Options Windows Help
from turtle import *
title('Prvi program')
# ovaj program će nacrtati pravokutnik stranica 100 i 50.

fd(100) # crtanje stranice duljine 100
rt(90) # okretanje za 90 stupnjeva
fd(50)
rt(90)
fd(100)
rt(90)
fd(50)
rt(90)
```

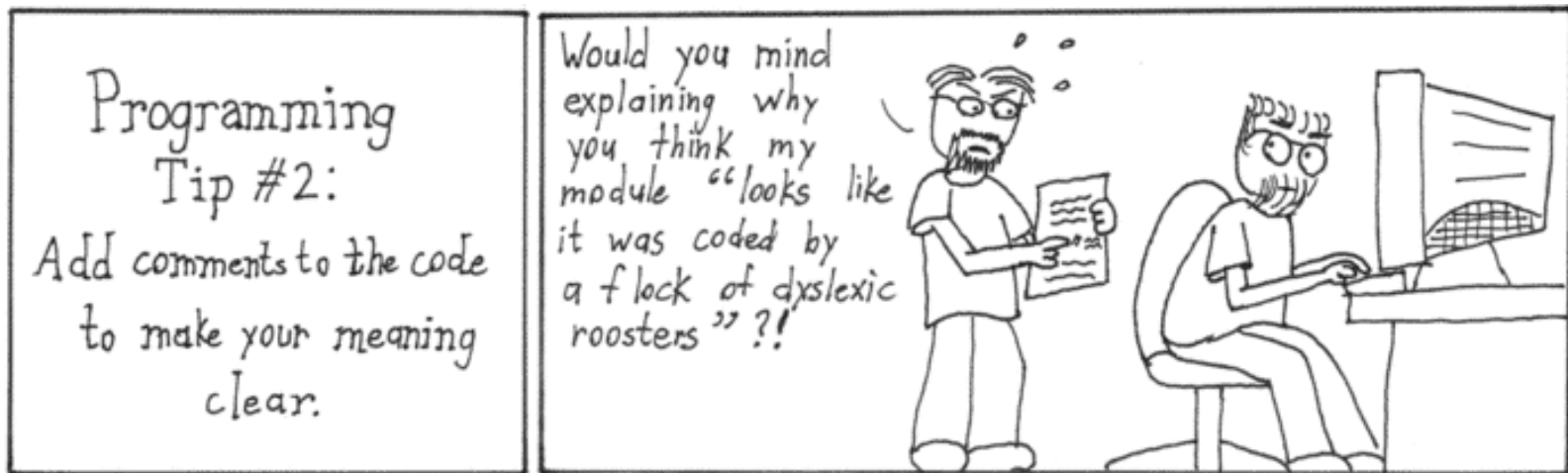


1. Napišite kod
2. Spremite modul
3. Pokrenite ga s F5

# Komentari

---

- ▶ Tekst koji počinje znakom # se neće izvršiti
- ▶ To su komentari programeru koji olakšavaju snalaženje u kodu





# Unos podataka

---

- ▶ `input()`
- ▶ Nakon pokretanja (**Run** ili **F5**), program čeka korisnika na upis nekog podatka
- ▶ Nakon unosa, korisnik mora pritisnuti **ENTER**
- ▶ Na temelju onoga što korisnik upiše, program vraća rezultat, spremljen u varijablu
  
- ▶ `ime_unesene_varijable = input('Tekst koji želimo ispisati na ekran prije unosa vrijednosti varijable: ')`

# Zadatak: kvadrat

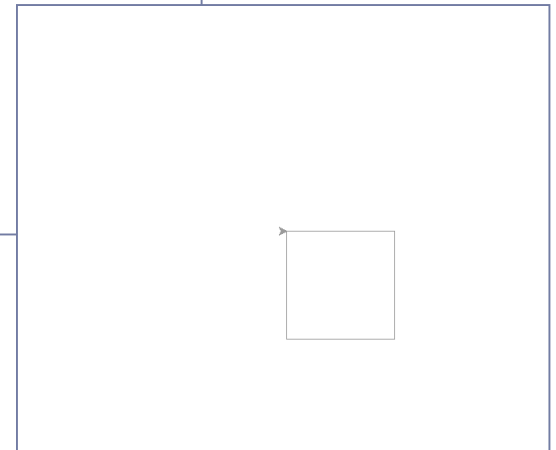
---

- ▶ Napiši program koji će korisnika tražiti da unese duljinu stranice kvadrata, a zatim će nacrtati kvadrat čija je stranica jednaka unesenom broju.

```
from turtle import *
title('Kvadrat')
# ovaj program će nacrtati kvadrat duljine stranice po izboru
duljina = input('Unesi duljinu stranice kvadrata: ')

fd(duljina) # crtanje stranice duljine duljina
rt(90)      # okretanje za 90 stupnjeva
fd(duljina)
rt(90)
fd(duljina)
rt(90)
fd(duljina)
rt(90)
```

```
>>>
Unesi duljinu stranice kvadrata: 134
```

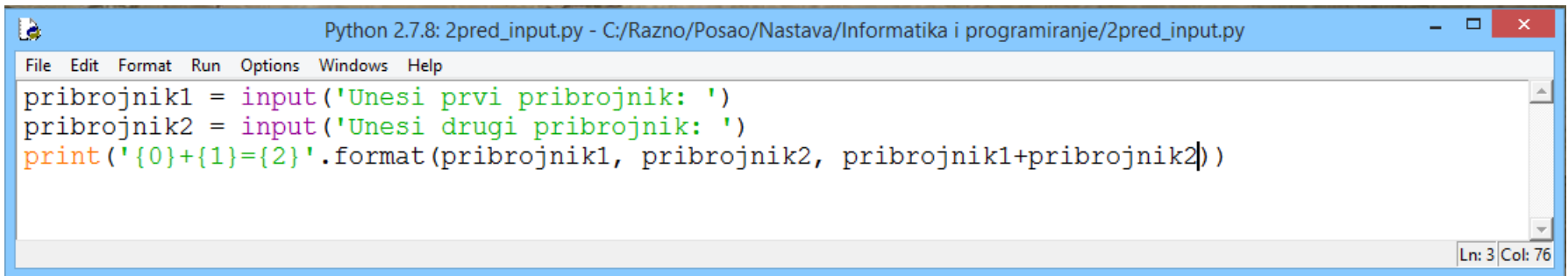


# Primjer: unos broja

---

- ▶ Napiši program koji će tražiti unos dva broja, a zatim će ispisati njihov zbroj.

```
>>>  
Unesi prvi pribrojnik: 5  
Unesi drugi pribrojnik: 6  
5+6=11  
>>>
```



```
Python 2.7.8: 2pred_input.py - C:/Razno/Posao/Nastava/Informatika i programiranje/2pred_input.py  
File Edit Format Run Options Windows Help  
pribrojnik1 = input('Unesi prvi pribrojnik: ')  
pribrojnik2 = input('Unesi drugi pribrojnik: ')  
print('{0}+{1}={2}'.format(pribrojnik1, pribrojnik2, pribrojnik1+pribrojnik2))  
Ln: 3 Col: 76
```

# Pisanje imena varijabli

---

- ▶ Slova, znamenke i podvlake
- ▶ Preporuka: prvi znak mora biti slovo
- ▶ Paziti na velika i mala slova
- ▶ Preporuka: ne koristiti hrvatske diakritičke znakove
- ▶ Primjeri imena: `a1`, `a_1`, `A1`, `A_1`, `vrijeme`, `nagib`
  
- ▶ Za lakše snalaženje – dulja imena, ime od više riječi
- ▶ Pojedine riječi odvojiti s `_`: `temperatura_zraka`
- ▶ ili napisati veliko slovo svake riječi: `TemperaturaVode`

# Ključne riječi u Pythonu

---

- ▶ Rezervirane riječi koje ne mogu biti ime našeg programa ili naših varijabli

<code>and</code>	<code>as</code>	<code>assert</code>	<code>break</code>	<code>class</code>
<code>continue</code>	<code>def</code>	<code>del</code>	<code>elif</code>	<code>else</code>
<code>except</code>	<code>exec</code>	<code>finally</code>	<code>for</code>	<code>from</code>
<code>global</code>	<code>if</code>	<code>import</code>	<code>in</code>	<code>is</code>
<code>lambda</code>	<code>not</code>	<code>or</code>	<code>pass</code>	<code>print</code>
<code>raise</code>	<code>return</code>	<code>try</code>	<code>while</code>	<code>with</code>
<code>yield</code>				

# Zadatak: input

---

- ▶ Strelica je izbačena vertikalno u vis početnom brzinom  $v$ . Napišite program koji će odrediti visinu strelice nakon  $t$  sekundi leta. Program treba tražiti unos početne brzine  $v$ , i vrijeme  $t$ , a ispisati visinu strelice.
- ▶ Podsjetnik: Ovdje se radi o vertikalnom hicu. Visina se određuje iz

$$y(t) = v \cdot t - g \cdot t^2 / 2$$

# Zadatak: input

---

- ▶ Ovdje se radi o vertikalnom hicu. Visina se određuje iz

$$y(t) = v \cdot t - g \cdot t^2 / 2$$

```
>>>  
Unesi pocetnu brzinu v i vrijeme leta t: 23.5, 3  
Visina strelice nakon 3 sekundi leta iznosi 26.4
```

```
v, t = input('Unesi pocetnu brzinu v i vrijeme leta t: ')  
print('Visina strelice nakon {0} sekundi leta iznosi {1}'.format(t, v*t-9.8*t**2/2))
```

# Ponavljanje

---

- ▶ Program za crtanje kvadrata četiri puta ponavlja iste naredbe

```
from turtle import *
title('Kvadrat')
# ovaj program će nacrtati kvadrat duljine stranice po izboru
duljina = input('Unesi duljinu stranice kvadrata: ')

fd(duljina) # crtanje stranice duljine duljina
rt(90)      # okretanje za 90 stupnjeva
fd(duljina)
rt(90)
fd(duljina)
rt(90)
fd(duljina)
rt(90)
```

- ▶ Možemo li to skratiti?

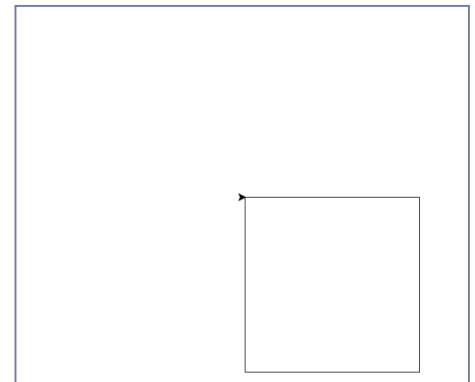


- 
- ▶ Možemo!
  - ▶ Koristeći PETLJU

```
from turtle import *
title('Kvadrat petlja')
# ovaj program će nacrtati kvadrat duljine stranice po izboru
# koristeći petlju
duljina = input('Unesi duljinu stranice kvadrata: ')

for i in range(4):
    fd(duljina) # crtanje stranice duljine duljina
    rt(90)     # okretanje za 90 stupnjeva
```

```
>>> |
Unesi duljinu stranice kvadrata: 250
```



I WILL USE 6006LE BEFORE ASKING DUMB QUESTIONS.  
I WILL USE 6006LE BEFORE ASKING DUMB QUESTIONS.  
I WILL USE 6006LE BEFORE ASKING DUMB QUESTIONS.  
I WILL USE 6006LE BEFORE ASKING DUMB QUESTIONS.  
I WILL USE 6006LE BEFORE ASKING DUMB QUESTIONS.  
I WILL USE 6006LE BEFORE ASKING DUMB QUESTIONS.  
I WILL USE 6006LE BEFORE ASKING DUMB QUESTIONS.  
I WILL USE 6006LE BEFORE ASKING DUMB QUESTIONS.  
I WILL USE 6006LE BEFORE ASKING DUMB QUESTIONS.  
I WILL USE 6006LE BEFORE ASKING DUMB QUESTIONS.  
I WILL USE 6006LE BEFORE ASKING DUMB QUESTIONS.  
I WILL USE 6006LE BEFORE ASKING DUMB QUESTIONS.  
I WILL USE 6006LE BEFORE ASKING DUMB QUESTIONS.  
I WILL USE 6006LE BEFORE ASKING DUMB QUESTIONS.  
I WILL USE 6006LE BEFORE ASKING DUMB QUESTIONS.



```
#include <stdio.h>
main ()
{
    int a;
    for (a = 0; a <= 100; a++)
        printf("I will search google before asking dumb questions");
    return 0;
}
```



# FOR petlja

```
for i in range(n):  
    naredba1  
    naredba2  
    naredba3
```

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

- ▶ `naredba1`, `naredba2` i `naredba3` se ponavljaju točno `n` puta
- ▶ ove tri naredbe zajedno čine `blok_naredbi`
- ▶ `blok_naredbi` se ponavlja točno `n` puta
- ▶ `blok_naredbi` započinje nakon dvotočke `:`
- ▶ `blok_naredbi` se piše uvučeno, obično 4 razmaka ili 1 tab
- ▶ nakon `n` ponavljanja – izvršava se **prva naredba nakon bloka** – ona se prepoznaje tako što **nije uvučena**

# FOR petlja

```
for i in range(n):  
    naredba1  
    naredba2  
    naredba3
```

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

- ▶ `blok_naredbi` se ponavlja `n` puta
- ▶ Prvi prolaz: `i=0`
- ▶ Drugi prolaz: `i=1`
- ▶ Zadnji prolaz: `i=n-1`
- ▶ Python počinje brojiti s **NULOM!**

# FOR petlja

```
for i in range(n):  
    naredba1  
    naredba2  
    naredba3
```

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

- ▶ Pseudojezik (kao program, ali slobodnim “jezikom”)

```
za i je 0 do n-1 ponavljaaj:  
    blok_naredbi
```

- ▶ Kada se izvrši `naredba3`, ponovno se izvršava `naredba1` – dolazi do ponavljanja, petlje.
- ▶ Ovakva vrsta petlje naziva se **FOR petlja**

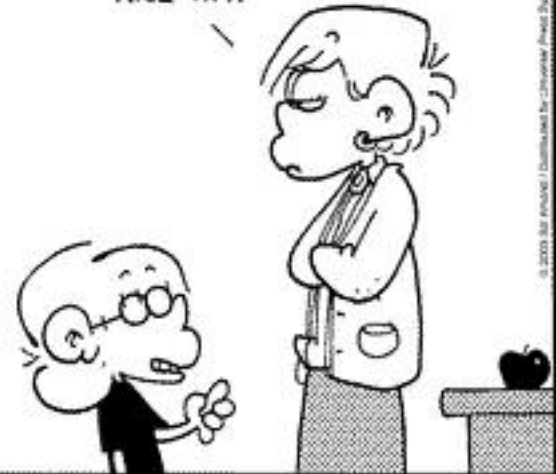
```
#include <stdio.h>
int main(void)
{
    int count;

    for (count = 1; count <= 500; count++)
        printf("I will not throw paper airplanes in class.");

    return 0;
}
```

AMEO 10-3

NICE TRY.



© 2008 All rights reserved. All trademarks are acknowledged. All rights reserved.

# Primjer – prirodni brojevi

---

- ▶ Napišite program koji će čekati unos jednog prirodnog broja **n**. Program treba ispisati na ekranu sve prirodne brojeve od **1** do **n**.

```
n = input('Unesi prirodni broj n: ')
# ispis svih prirodnih brojeva do n
for i in range(n):
    print('{0}'.format(i+1))
```

```
>>>
Unesi prirodni broj n: 4
1
2
3
4
```

# Primjer – neparni brojevi

---

- ▶ Napišite program koji će čekati unos jednog prirodnog broja **n**. Program treba ispisati na ekranu sve neparne brojeve od **1** do **n**.

```
n = input('Unesi prirodni broj n: ')
# ispis svakog drugog broja do n
for i in range(0, n, 2):
    print('{0}'.format(i+1))
```

```
>>>
Unesi prirodni broj n: 15
1
3
5
7
9
11
13
15
```

- ▶ FOR petlja “na preskokce”

```
for i in range(pocetak, zadnji_broj-1, korak):
    blok_naredbi
```



# Zadaci za vježbu

---

- ▶ Napišite program koji će tražiti unos dva broja i ispisati njihovu sumu.
- ▶ Napišite program koji će tražiti unos dva broja i ispisati sumu svih brojeva između unesenih brojeva – uključujući i njih.

# Primjer - faktorijela

---

- ▶ Napišite program koji će tražiti korisnika prirodni broj  $N$ , te odrediti vrijednost  $N!$  Ispisujte i međurezultate.
- ▶ Podsjetnik:  $N! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot N$ .

# Primjer – faktorijela

---

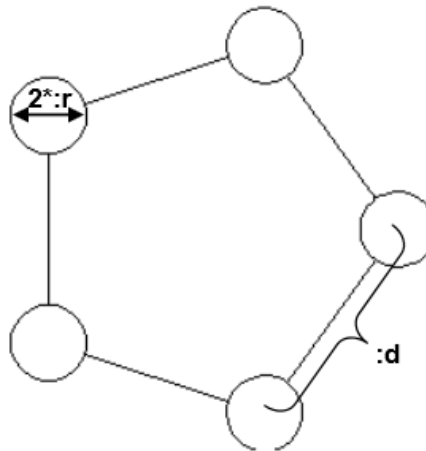
```
N = input('Unesi prirodni broj N: ')
faktorijela = 1
for i in range(N):
    faktorijela = faktorijela * (i+1)
    print('{0} --> {1}'.format(i+1, faktorijela))
print('Faktorijela zadanog broja {0} je: {1}'.format(N, faktorijela))
```

```
>>> ===== RESTART =====
>>>
Unesi prirodni broj N: 4
1 --> 1
2 --> 2
3 --> 6
4 --> 24
Faktorijela zadanog broja 4 je: 24
```

# Primjer - dvorac

---

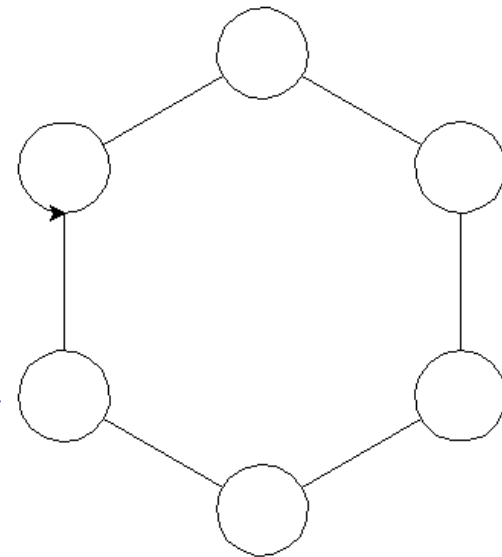
- ▶ Napiši program koji će crtati tlocrt srednjovjekovnog dvorca. Dvorac je pravilni  $n$ -terokut duljine stranice  $d$ . Svaka strana završava kulom i stražarnicom radijusa  $r$ . Crta koja predstavlja zid ne smije ulaziti u unutrašnjost kružnice koja predstavlja kulu.
- ▶ Pri pokretanju programa, program mora tražiti unos broja stranica dvorca  $n$ , duljinu stranice dvorca  $d$ , te radijus kule  $r$ .



# Primjer - dvorac

```
from turtle import *
title('Dvorac')
# ovaj program će nacrtati dvorac koristeći petlju
n, d, r = input('Unesi broj stranica dvorca, duljinu zida, radijus kule: ')

for i in range(n):
    circle(r) # nacrtaj kulu
    pu()     # digni olovku
    lt(90)   # okreni se lijevo za 90°
    fd(r)    # hodaj ravno udaljenost r
    rt(360/n) # usmjeri se u smjeru novog zida
    fd(r)    # hodaj ravno r
    pd()     # spusti olovku
    fd(d-2*r) # nacrtaj zid
    rt(90)   # okreni se za 90° zbog crtanja kule
```



```
Unesi broj stranica dvorca, duljinu zida, radijus kule: 6,150,30
>>>
```

# Donošenje odluka

---

- ▶ Napiši program koji će učitati visine Marije i Ivana i ispisati tko je viši.

```
m, i = input('Unesi visinu Marije i Ivana: ')  
  
if m<i:  
    print('Ivan je visi.')  
else:  
    print('Marija je visa.')
```

```
>>>  
Unesi visinu Marije i Ivana: 165,195  
Ivan je visi.
```

# IF

---

```
if uvjet:  
    naredba1_1  
    naredba1_2  
    naredba1_3
```

```
if uvjet:  
    blok_naredbi_1
```

- ▶ `naredba1_1`, `naredba1_2` i `naredba1_3` se izvršavaju ako je ispunjen `uvjet`
- ▶ ove tri naredbe zajedno čine `blok_naredbi_1`
- ▶ `blok_naredbi_1` se izvršava ako je ispunjen `uvjet`
- ▶ `blok_naredbi_1` započinje nakon dvotočke `:`
- ▶ `blok_naredbi_1` se piše uvučeno, 4 razmaka ili `| tab`

# IF ELSE

---

```
if uvjet:
    naredba1_1
    naredba1_2
    naredba1_3
else:
    naredba2_1
    naredba2_2
    naredba2_3
```

```
if uvjet:
    blok_naredbi_1
else:
    blok_naredbi_2
```

- ▶ naredba1\_1, naredba1\_2 i naredba1\_3 se izvršavaju ako **JE** ispunjen uvjet
- ▶ blok\_naredbi\_1 se izvršava ako **JE** ispunjen uvjet
- ▶ naredba2\_1, naredba2\_2 i naredba2\_3 se izvršavaju ako **NIJE** ispunjen uvjet
- ▶ blok\_naredbi\_2 se izvršava ako **NIJE** ispunjen uvjet



# Grananje ili račvanje

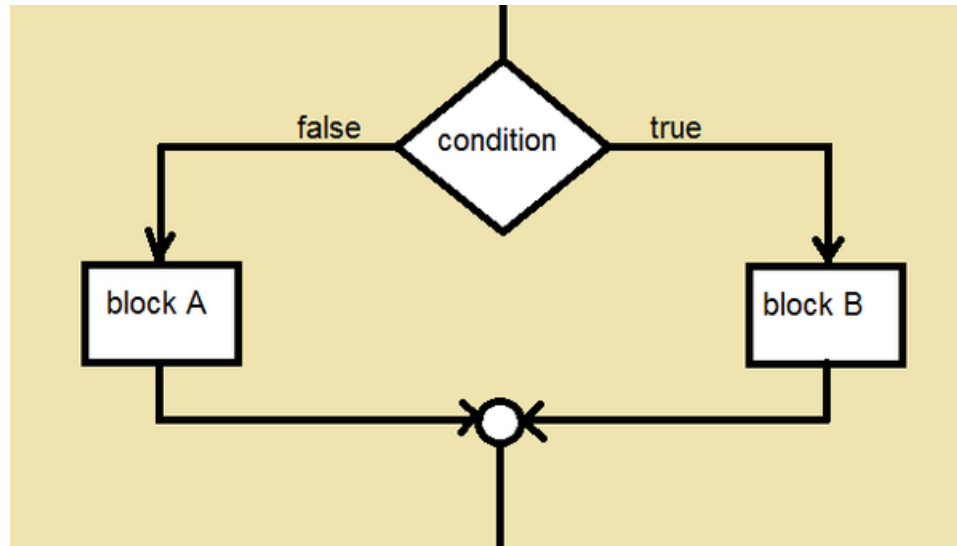
---

```
if uvjet:  
    blok_naredbi_1  
else:  
    blok_naredbi_2
```

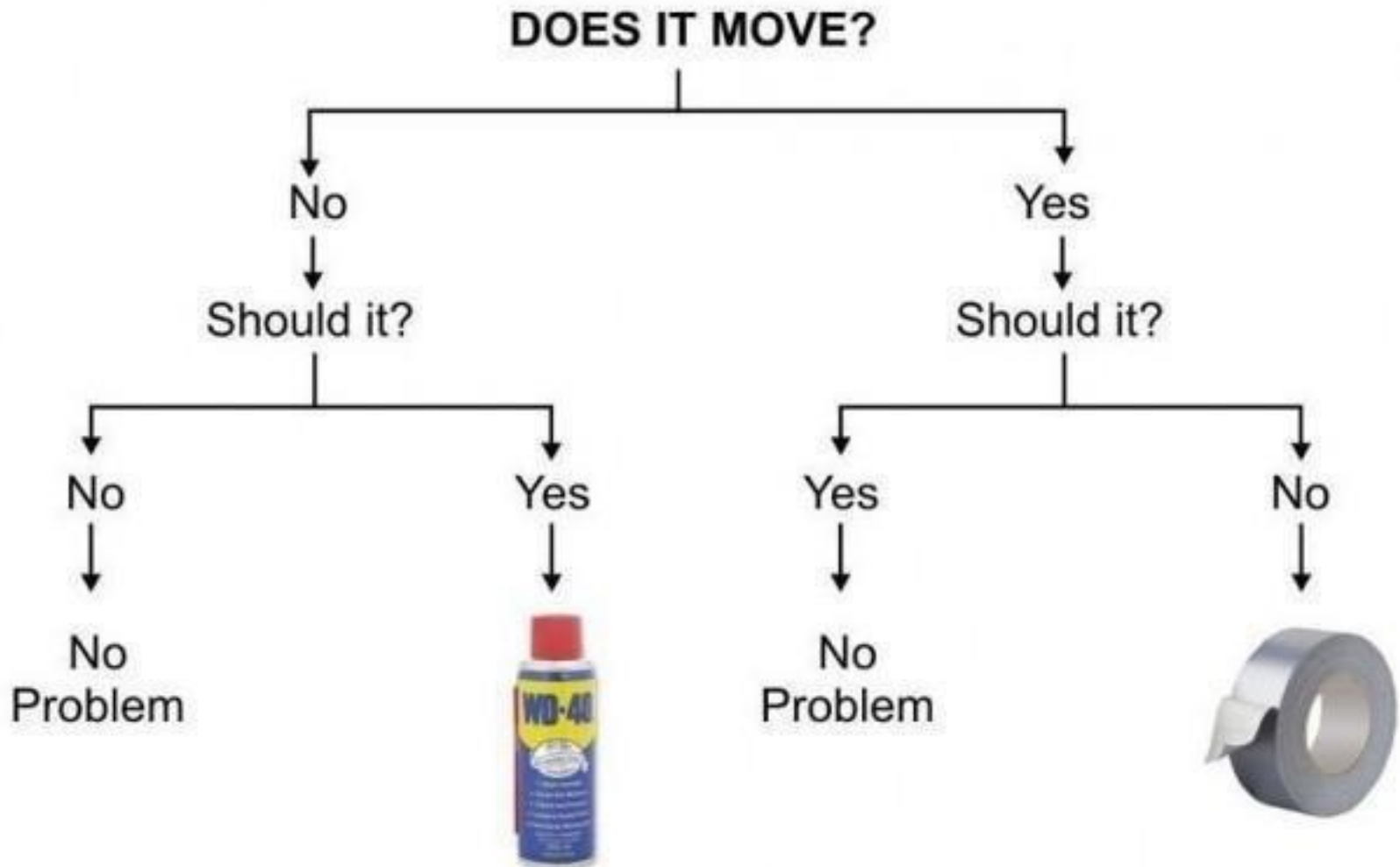
- ▶ Pseudojezik (kao program, ali slobodnim “jezikom”)

```
ako je uvjet onda:  
    blok_naredbi_1  
inače:  
    blok_naredbi_2
```

- ▶ Ako **JE** uvjet ispunjen izvršava se blok\_naredbi\_1, a ako uvjet **NIJE** ispunjen izvršava se blok\_naredbi\_2
- ▶ Program donosi odluku – izabire alternativni blok naredbi – program se **grana** ili **račva**



# Engineering Flowchart



# AM I A HORSE?

A HELPFUL FLOW CHART



# Primjer: paran broj

- ▶ Napišite program koji će za učitani broj reći je li paran ili neparan.

```
broj = input('Unesi broj: ')
if broj%2==1:
    print('{0} je neparan.'.format(broj))
else:
    print('{0} je paran.'.format(broj))
```

Broj je neparan ako mu je ostatak pri dijeljenju sa 2 jednak 1

Broj je paran ako mu je ostatak pri dijeljenju sa 2 jednak 0

```
>>> ===== RESTART =====
>>>
Unesi broj: 45
45 je neparan.
>>> ===== RESTART =====
>>>
Unesi broj: 24
24 je paran.
```

# Više o uvjetu

---

- ▶ Uvjet je ispunjen = logični izraz je istinit = `True`
- ▶ Uvjet nije ispunjen = logični izraz je lažan = `False`
  
- ▶ Relacijski operatori
- ▶ Logički operatori

# Relacijski operatori

---

- ▶ Operatori uspoređivanja
- ▶ Rezultat usporedbe je `True` ili `False`

Operator	Značenje
>	Veće od
<	Manje od
>=	Veće ili jednako
<=	Manje ili jednako
==	Jednako
!=	Nije jednako, različito

- ▶ **Pazi!**
  - ▶ Provjera je li nešto jednako radi se s dva znaka jednakosti
  - ▶ Jedan znak jednakosti se koristi za pridruživanje

# Primjer: relacijski operatori

```
>>> 12>18
False
>>> 12<18
True
>>> 12>=18
False
>>> 12<=18
True
>>> 12==18
False
>>> 12!=18
True
>>> 12=18
SyntaxError: can't assign to literal
```

GOOD CODERS...



... KNOW WHAT THEY'RE DOING



# Primjer: škola

---

- ▶ Napiši program koji će za unesenu starost djeteta reći ide li u osnovnu školu.
- ▶ Pretpostavimo:
  - ▶ U osnovnu školu idu djeca između 7 i 14 godina.

```
ako je dijete_staro >= 7 onda:  
    print('Dijete ide u osnovnu školu')
```



```
ako je dijete_staro <= 14 onda:  
    print('Dijete ide u osnovnu školu')
```



```
ako je dijete_staro >= 7 i dijete_staro <= 14 onda:  
    print('Dijete ide u osnovnu školu')
```



# Logički operatori

---

- ▶ Ako nam jedan relacijski operator nije dovoljan za konačnu odluku
- ▶ Logički operatori se koriste za dobivanje složenijih logičkih izraza
- ▶ I / ILI / NE

# Logički operatori

Operator	Operacija	Ponašanje																								
<b>and</b>	<p>I operacija Konjunkcija</p> <p>ISTINA – samo ako <b>oba</b> operanda istinita</p>	<table border="1"> <tr> <td>False</td> <td>False</td> <td>False</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>False</td> <td>True</td> <td>False</td> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>True</td> <td>False</td> <td>False</td> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>True</td> <td>True</td> <td>True</td> <td>1</td> <td>1</td> <td>1</td> </tr> </table>	False	False	False	0	0	0	False	True	False	0	1	0	True	False	False	1	0	0	True	True	True	1	1	1
False	False	False	0	0	0																					
False	True	False	0	1	0																					
True	False	False	1	0	0																					
True	True	True	1	1	1																					
<b>or</b>	<p>II operacija Disjunkcija</p> <p>ISTINA – ako <b>bar jedan</b> operand istinit</p>	<table border="1"> <tr> <td>False</td> <td>False</td> <td>False</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>False</td> <td>True</td> <td>True</td> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>True</td> <td>False</td> <td>True</td> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>True</td> <td>True</td> <td>True</td> <td>1</td> <td>1</td> <td>1</td> </tr> </table>	False	False	False	0	0	0	False	True	True	0	1	1	True	False	True	1	0	1	True	True	True	1	1	1
False	False	False	0	0	0																					
False	True	True	0	1	1																					
True	False	True	1	0	1																					
True	True	True	1	1	1																					
<b>not</b>	<p>NE operacija Komplementiranje</p> <p>ISTINA – ako je operand <b>bio lažan</b></p>	<table border="1"> <tr> <td>False</td> <td>True</td> <td>0</td> <td>1</td> </tr> <tr> <td>True</td> <td>False</td> <td>1</td> <td>0</td> </tr> </table>	False	True	0	1	True	False	1	0																
False	True	0	1																							
True	False	1	0																							

# Redoslijed operacija

---

## 1. Aritmetičke operacije

1. Potenciranje (\*\*)
2. Negacija (-)
3. Množenje, dijeljenje, ostatak (\*, /, %)
4. Zbrajanje, oduzimanje (+, -)

## 2. Relacijske operacije (>, <, >=, <=, ==, !=)

## 3. Logičke operacije

1. Negacija (not)
2. I operacija (and)
3. Ili operacija (or)

# Primjer: škola (if)

- ▶ Napiši program koji će za unesenu starost djeteta reći ide li u osnovnu školu.
  - ▶ U osnovnu školu idu djeca između 7 i 14 godina.

```
dijete_staro = input('Unesi starost djeteta: ')
if dijete_staro >= 7 and dijete_staro <= 14:
    print('Dijete ide u osnovnu školu.')
```

```
>>> ===== RESTART =====
>>>
Unesi starost djeteta: 16
>>> ===== RESTART =====
>>>
Unesi starost djeteta: 13
Dijete ide u osnovnu školu.
>>> ===== RESTART =====
>>>
Unesi starost djeteta: 5
>>> ===== RESTART =====
>>>
Unesi starost djeteta: 7
Dijete ide u osnovnu školu.
>>> ===== RESTART =====
>>>
Unesi starost djeteta: 14
Dijete ide u osnovnu školu.
```

# Primjer: škola (if-else)

---

- ▶ Napiši program koji će za unesenu starost djeteta reći ide li u osnovnu školu ili ne ide u osnovnu školu.
  - ▶ U osnovnu školu idu djeca između 7 i 14 godina.

```
dijete_staro = input('Unesi starost djeteta: ')
if dijete_staro >= 7 and dijete_staro <= 14:
    print('Dijete ide u osnovnu školu.')
else:
    print('Dijete ne ide u osnovnu školu.')
```

```
>>> ===== RESTART =====
>>>
Unesi starost djeteta: 5
Dijete ne ide u osnovnu školu.
>>> ===== RESTART =====
>>>
Unesi starost djeteta: 12
Dijete ide u osnovnu školu.
>>> ===== RESTART =====
>>>
Unesi starost djeteta: 16
Dijete ne ide u osnovnu školu.
```

# Primjer: škola (suprotni operator)

- ▶ Napiši program koji će za unesenu starost djeteta reći ide li u osnovnu školu ili ne ide u osnovnu školu.
  - ▶ U osnovnu školu idu djeca između 7 i 14 godina.

<pre>dijete_staro = input('Unesi starost djeteta: ') if dijete_staro &gt;= 7 and dijete_staro &lt;= 14:     print('Dijete ide u osnovnu skolu.') else:     print('Dijete ne ide u osnovnu skolu.')</pre>	<pre>dijete_staro = input('Unesi starost djeteta: ') if dijete_staro &lt; 7 or dijete_staro &gt; 14:     print('Dijete ne ide u osnovnu skolu.') else:     print('Dijete ide u osnovnu skolu.')</pre>
--	---

```
>>> ===== RESTART =====
>>>
Unesi starost djeteta: 5
Dijete ne ide u osnovnu skolu.
>>> ===== RESTART =====
>>>
Unesi starost djeteta: 12
Dijete ide u osnovnu skolu.
>>> ===== RESTART =====
>>>
Unesi starost djeteta: 16
Dijete ne ide u osnovnu skolu.
```

# Suprotni operator

---

Operator	Suprotni operator
$>$	$\leq$
$<$	$\geq$
$\geq$	$<$
$\leq$	$>$
$==$	$\neq$
$\neq$	$==$

## ► Primjer:

Operacija	Primjer	Brojevi koji nisu uključeni	Suprotna operacija
$x > 3$	$x = 4, 5, 6, \dots$	$x = 3, 2, 1, \dots$	$x \leq 3$
$x < 6$	$x = 5, 4, 3, \dots$	$x = 6, 7, 8, \dots$	$x \geq 6$
$x == 7$	$x = 7$	$x = \dots, 5, 6, 8, 9, \dots$	$x \neq 7$



# Primjer: škola (if – elif – else)

---

- ▶ Napiši program koji će za unesenu starost djeteta reći ide li u osnovnu ili srednju školu.
- ▶ Pretpostavimo:
  - ▶ U osnovnu školu idu djeca između 7 i 14 godina,
  - ▶ U srednju školu idu djeca između 15 i 18 godina.

```
ako je dijete_staro >= 7 i dijete_staro <= 14 onda:  
    print('Dijete ide u osnovnu školu.')
```

```
inače ako je dijete_staro >= 15 i dijete_staro <= 18 onda:  
    print('Dijete ide u srednju školu.')
```

```
inače:  
    print('Dijete ne ide u školu.')
```

# IF – ELIF – ELSE

---

## ▶ Pseudojezik

```
ako je uvjet_1 onda:
    blok_naredbi_1
inače ako je uvjet_2 onda:
    blok_naredbi_2
...
inače ako je uvjet_N onda:
    blok_naredbi_N
inače:
    blok_naredbi
```

## Python

```
if uvjet_1:
    blok_naredbi_1
elif uvjet_2:
    blok_naredbi_2
...
elif uvjet_N:
    blok_naredbi_N
else:
    blok_naredbi
```

- ▶ **Ako je** uvjet\_1 ispunjen izvršava se blok\_naredbi\_1,
- ▶ **Ako uvjet\_1 nije** ispunjen, ali je ispunjen uvjet\_2 izvršava se blok\_naredbi\_2,
- ▶ ...
- ▶ **Ako nije ispunjen niti jedan** od uvjeta uvjet\_1,... uvjet\_N, izvršava se blok\_naredbi

# Primjer: škola (if – elif – else)

---

- ▶ Napiši program koji će za unesenu starost djeteta reći ide li u osnovnu ili srednju školu.
  - ▶ U osnovnu školu idu djeca između 7 i 14 godina,
  - ▶ U srednju školu idu djeca između 15 i 18 godina.

```
dijete_staro = input('Unesi starost djeteta: ')

if dijete_staro >= 7 and dijete_staro <= 14:
    print('Dijete ide u osnovnu školu.')
elif dijete_staro >= 15 and dijete_staro <= 18:
    print('Dijete ide u srednju školu.')
else:
    print('Dijete ne ide u školu.')
```

```
>>> =====
>>>
Unesi starost djeteta: 4
Dijete ne ide u školu.
>>> =====
>>>
Unesi starost djeteta: 8
Dijete ide u osnovnu školu.
>>> =====
>>>
Unesi starost djeteta: 15
Dijete ide u srednju školu.
>>> =====
>>>
Unesi starost djeteta: 19
Dijete ne ide u školu.
```

# Primjer: škola (if – elif – else)

- ▶ Napiši program koji će za unesenu starost osobe reći ide li u osnovnu ili srednju školu ili na fakultet.
  - ▶ U osnovnu školu idu djeca između 7 i 14 godina,
  - ▶ U srednju školu idu djeca između 15 i 18 godina,
  - ▶ Na fakultet idu ljudi između 19 i 23 godine.

```
dijete_staro = input('Unesi starost osobe: ')

if dijete_staro >= 7 and dijete_staro <= 14:
    print('Osoba ide u osnovnu skolu.')
elif dijete_staro >= 15 and dijete_staro <= 18:
    print('Osoba ide u srednju skolu.')
elif dijete_staro >= 19 and dijete_staro <= 23:
    print('Osoba ide na fakultet.')
else:
    print('Osoba ne ide ni u skolu, ni na fakultet.')
```

```
>>> ===== RESTART
>>>
Unesi starost osobe: 6
Osoba ne ide ni u skolu, ni na fakultet.
>>> ===== RESTART
>>>
Unesi starost osobe: 8
Osoba ide u osnovnu skolu.
>>> ===== RESTART
>>>
Unesi starost osobe: 15
Osoba ide u srednju skolu.
>>> ===== RESTART
>>>
Unesi starost osobe: 19
Osoba ide na fakultet.
>>> ===== RESTART
>>>
Unesi starost osobe: 24
Osoba ne ide ni u skolu, ni na fakultet.
```

# Primjer: škola (if – elif – else – for)

---

- ▶ Napiši program koji će za unesenu starost osobe reći ide li u osnovnu ili srednju školu ili na fakultet.
- ▶ Program mora tražiti unos godina za 5 osoba.
  - ▶ U osnovnu školu idu djeca između 7 i 14 godina,
  - ▶ U srednju školu idu djeca između 15 i 18 godina,
  - ▶ Na fakultet idu ljudi između 19 i 23 godine.

```
for i in range(5):
    dijete_staro = input('Unesi starost osobe: ')

    if dijete_staro >= 7 and dijete_staro <= 14:
        print('Osoba ide u osnovnu školu.')
    elif dijete_staro >= 15 and dijete_staro <= 18:
        print('Osoba ide u srednju školu.')
    elif dijete_staro >= 19 and dijete_staro <= 23:
        print('Osoba ide na fakultet.')
    else:
        print('Osoba ne ide ni u školu, ni na fakultet.')
```

```
>>>
Unesi starost osobe: 4
Osoba ne ide ni u školu, ni na fakultet.
Unesi starost osobe: 9
Osoba ide u osnovnu školu.
Unesi starost osobe: 17
Osoba ide u srednju školu.
Unesi starost osobe: 21
Osoba ide na fakultet.
Unesi starost osobe: 26
Osoba ne ide ni u školu, ni na fakultet.
```

# Primjer: kružnice u boji

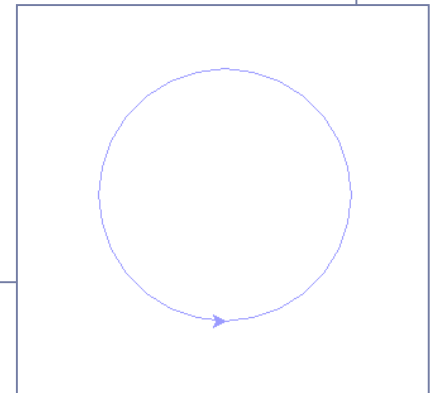
---

- ▶ Napiši program koji će čekati unos boje i radijusa kružnice, te ju nacrtati.

```
from turtle import *
title('Kružnica u boji')

boja = input('Izaberi boju kružnice (1 - crveno, 2 - plavo, 3 - zeleno): ')
radijus = input('Unesi radijus kružnice: ')

if boja == 1:
    color('red'); circle(radijus)
elif boja == 2:
    color('blue'); circle(radijus)
elif boja == 3:
    color('green'); circle(radijus)
```



```
>>>
Izaberi boju kružnice (1 - crveno, 2 - plavo, 3 - zeleno): 2
Unesi radijus kružnice: 100
>>>
```

# Primjer: parni brojevi (1)

---

- ▶ Napiši program koji će tražiti unos dva prirodna broja i ispisati sve parne brojeve između njih.
- ▶ Npr. Ulaz: 45, 51. Izlaz: 46, 48, 50.

```
a, b = input('Unesi dva prirodna broja: ')
print('Parni brojevi između {0} i {1} su: '.format(a,b))
for i in range(a, b, 1):
    if not(i%2):
        print('{0}'.format(i))
```

```
>>>
Unesi dva prirodna broja: 45,51
Parni brojevi između 45 i 51 su:
46
48
50
```

- ▶ A što ako je prvi broj veći od drugoga?

```
>>>
Unesi dva prirodna broja: 51,45
Parni brojevi između 51 i 45 su:
```

## Primjer: parni brojevi (2)

---

- ▶ Napiši program koji će tražiti unos dva prirodna broja i ispisati sve parne brojeve između njih.
- ▶ Npr. Ulaz: 45, 51. Izlaz: 46, 48, 50.

```
a, b = input('Unesi dva prirodna broja: ')
print('Parni brojevi između {0} i {1} su: '.format(a,b))
if a<=b:
    for i in range(a, b, 1):
        if not(i%2):
            print('{0}'.format(i))
else:
    for i in range(b, a, 1):
        if not(i%2):
            print('{0}'.format(i))
```

```
>>> =====
>>>
Unesi dva prirodna broja: 45,51
Parni brojevi između 45 i 51 su:
46
48
50
>>> =====
>>>
Unesi dva prirodna broja: 51,45
Parni brojevi između 51 i 45 su:
46
48
50
```



# Primjer: broj parnih brojeva (1)

- ▶ Napiši program koji će tražiti unos dva prirodna broja i ispisati koliko ima parnih brojeva između njih.
- ▶ Npr. Ulaz: 45, 51. Izlaz: 3.

```
a, b = input('Unesi dva prirodna broja: ')
parni = 0
if a<=b:
    for i in range(a, b, 1):
        if not(i%2):
            parni = parni + 1
else:
    for i in range(b, a, 1):
        if not(i%2):
            parni = parni + 1
print('Između {0} i {1} ima {2} parnih brojeva.'.format(a,b,parni))
```

brojilo parnih brojeva

na početku ga je  
potrebno postaviti na 0  
= **INICIJALIZACIJA**

tokom programa se  
njegova vrijednost  
povećava za 1

```
>>> =====
>>>
Unesi dva prirodna broja: 45,51
Između 45 i 51 ima 3 parnih brojeva.
>>> =====
>>>
Unesi dva prirodna broja: 51,45
Između 51 i 45 ima 3 parnih brojeva.
```

# Primjer: broj parnih brojeva (2)

- ▶ Napiši program koji će tražiti unos dva prirodna broja i ispisati koliko ima parnih brojeva između njih.
- ▶ Npr. Ulaz: 45, 51. Izlaz: 3.

```
a, b = input('Unesi dva prirodna broja: ')
parni = 0
if a<=b:
    for i in range(a, b, 1):
        if not(i%2):
            parni = parni + 1
else:
    for i in range(b, a, 1):
        if not(i%2):
            parni = parni + 1
print('Između {0} i {1} ima {2} parnih brojeva.'.format(a,b,parni))
```

brojilo parnih brojeva na početku ga je potrebno postaviti na 0

tokom programa se njegova vrijednost povećava za 1

```
>>> =====
>>>
Unesi dva prirodna broja: 45,51
Između 45 i 51 ima 3 parnih brojeva.
>>> =====
>>>
Unesi dva prirodna broja: 51,45
Između 51 i 45 ima 3 parnih brojeva.
```

# Operatori proširenog pridruživanja

---

- ▶ Povećavanje sadržaja nekog brojila
- ▶ Umjesto pisanja

```
brojac = brojac + 1
```

može se pisati

```
brojac += 1
```

- ▶ Ovakvo pridruživanje vrijednosti naziva se **prošireno pridruživanje**
- ▶ Moguće ga je provesti za sve aritmetičke operacije:  

```
+=, -=, *=, /=, //, %=, **=
```
- ▶ Pojednostavljuje pisanje program i ubrzava izvođenje

# Prošireno pridruživanje

---

1. Inicijalizirajte varijablu na početku programa ili prije operacije proširenog pridruživanja (npr. `parni=0`)
2. Prvi operand – vrijednost varijable koja stoji s lijeve strane znaka pridruživanja (npr. `parni`)
3. Drugi operand – broj ili varijabla koja stoji s desne strane znaka pridruživanja (npr. `1`)
4. Operacija – oznaka pored znaka jednakosti (`+`, `-`, `*`, `/`, `//`, `%`, `**`) (npr. `+`)
5. Nova vrijednost se pridružuje varijabli s lijeve strane znaka pridruživanja (npr. `parni=1`)

# Primjer: prošireno pridruživanje

---

- ▶ Što će ispisati sljedeći programi?

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

```
izlaz = 0
for i in range(6):
    izlaz += 2 * i
print(izlaz)
```

```
izlaz = 54
for i in range(3):
    izlaz //= 2
print(izlaz)
```

```
izlaz = 54
for i in range(3):
    izlaz //= 2
print(izlaz)
```

# Primjer: škola (if – elif – else – for)

---

- ▶ Napiši program koji će za unesenu starost osobe reći ide li u osnovnu ili srednju školu ili na fakultet.
- ▶ Program mora tražiti unos godina **za 5 osoba**.
  - ▶ U osnovnu školu idu djeca između 7 i 14 godina,
  - ▶ U srednju školu idu djeca između 15 i 18 godina,
  - ▶ Na fakultet idu ljudi između 19 i 23 godine.

```
for i in range(5):
    dijete_staro = input('Unesi starost osobe: ')

    if dijete_staro >= 7 and dijete_staro <= 14:
        print('Osoba ide u osnovnu školu.')
    elif dijete_staro >= 15 and dijete_staro <= 18:
        print('Osoba ide u srednju školu.')
    elif dijete_staro >= 19 and dijete_staro <= 23:
        print('Osoba ide na fakultet.')
    else:
        print('Osoba ne ide ni u školu, ni na fakultet.')
```

```
>>>
Unesi starost osobe: 4
Osoba ne ide ni u školu, ni na fakultet.
Unesi starost osobe: 9
Osoba ide u osnovnu školu.
Unesi starost osobe: 17
Osoba ide u srednju školu.
Unesi starost osobe: 21
Osoba ide na fakultet.
Unesi starost osobe: 26
Osoba ne ide ni u školu, ni na fakultet.
```

# Primjer: škola (if – elif – else – ?)

- ▶ Napiši program koji će za unesenu starost osobe reći ide li u osnovnu ili srednju školu ili na fakultet.
- ▶ Program mora tražiti unos godina **sve dok se ne unese negativna vrijednost**.
  - ▶ U osnovnu školu idu djeca između 7 i 14 godina,
  - ▶ U srednju školu idu djeca između 15 i 18 godina,
  - ▶ Na fakultet idu ljudi između 19 i 23 godine.

```
for i in range(5):  
    dijete_staro = input('Unesi starost osobe: ')  
  
    if dijete_staro >= 7 and dijete_staro <= 14:  
        print('Osoba ide u osnovnu skolu.')  
    elif dijete_staro >= 15 and dijete_staro <= 18:  
        print('Osoba ide u srednju skolu.')  
    elif dijete_staro >= 19 and dijete_staro <= 23:  
        print('Osoba ide na fakultet.')  
    else:  
        print('Osoba ne ide ni u skolu, ni na fakultet.')
```

---

▶ **for naredba**

- ▶ Broj ponavljanja naredbi iza `for` je unaprijed zadan

```
Za i je 0 do n-1 ponavljaaj:  
    blok_naredbi
```

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

- ▶ trebamo nešto slično `for`, ali što će se ponavljati **sve dok je ispunjen neki uvjet**

```
dok je uvjet čini:  
    blok_naredbi
```



# While petlja

---

```
while uvjet:  
    blok_naredbi
```

- ▶ `uvjet` – logički izraz (kao i u `if-elif-else`)
- ▶ Ako je logički izraz istinit (`True`) – uvjet ispunjen – `blok_naredbi` se izvršava
- ▶ Unutar `blok_naredbi` mora biti neka naredba koja utječe na `uvjet` kako bi se on promijenio u `False`
- ▶ Kada uvjet postane `False` (logički izraz postane lažan) – `blok_naredbi` se više neće izvršiti
- ▶ Python prelazi na prvu naredbu nakon petlje
- ▶ Ovakva vrsta petlje je **WHILE** petlja

# For – while (1)

---

- ▶ Što će ispisati program:

```
for i in range(10):  
    print(i),
```

- ▶ Koja je prva, a koja posljednja vrijednost ispisana na ekranu?
- ▶ Kod `while` petlje početnu vrijednost varijable `i` je potrebno inicijalizirati – zadati prije početka petlje

```
i = 0
```

- ▶ Petlja se ponavlja sve dok je `i < 10` – to je uvjet petlje
- ▶ Varijabla mora mijenjati svoju vrijednost svakim prolaskom kroz petlju kako bi uvjet postao lažan

```
i = i + 1    ili    i += 1
```

# For – while (2)

---

## ▶ For petlja

```
for i in range(10):  
    print(i),
```

## ▶ While petlja

```
i = 0  
while i < 10:  
    print(i),  
    i += 1
```

## ▶ Ispis:

```
>>>  
0 1 2 3 4 5 6 7 8 9
```

- ▶ Zarez iza naredbe `print` – kako bi se u sljedećem prolasku kroz petlju brojevi ispisivali u isti redak

# While petlja – česte pogreške

---

## ▶ For petlja

```
for i in range(10):  
    print(i),
```

## While petlja

```
i = 0  
while i<10:  
    print(i),  
    i += 1
```

## Ispis

```
>>>  
0 1 2 3 4 5 6 7 8 9
```

## ▶ Bez inicijalizacije

- ▶ U varijabli `i` može biti spremljen neki bezvezni broj → petlja se možda neće ni pokrenuti

```
while i<10:  
    print(i),  
    i += 1
```

## ▶ Beskonačna petlja

- ▶ Varijabla `i` se ne mijenja → uvjet je uvijek ispunjen → nasilno zaustavljanje programa tipkama CTRL+c

```
i = 0  
while i<10:  
    print(i),
```

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0  
Traceback (most recent call last):  
  File "F:/Documents/nastava/Informatika i programi  
ranje/2pred_for_while.py", line 24, in <module>  
    print(i),  
  File "C:\Python27\lib\idlelib\PyShell.py", line 1  
356, in write  
    return self.shell.write(s, self.tags)  
KeyboardInterrupt
```

# Primjer: škola (if – elif – else – while)

- ▶ Napiši program koji će za unesenu starost osobe reći ide li u osnovnu ili srednju školu ili na fakultet.
- ▶ Program mora tražiti unos godina **za 5 osoba**.
  - ▶ U osnovnu školu idu djeca između 7 i 14 godina,
  - ▶ U srednju školu idu djeca između 15 i 18 godina,
  - ▶ Na fakultet idu ljudi između 19 i 23 godine.

```
for i in range(5):
    dijete_staro = input('Unesi starost osobe: ')

    while i < 5:
        dijete_staro = input('Unesi starost osobe: ')

        if dijete_staro >= 7 and dijete_staro <= 14:
            print('Osoba ide u osnovnu školu.')
        elif dijete_staro >= 15 and dijete_staro <= 18:
            print('Osoba ide u srednju školu.')
        elif dijete_staro >= 19 and dijete_staro <= 23:
            print('Osoba ide na fakultet.')
        else:
            print('Osoba ne ide ni u školu, ni na fakultet.')

        i += 1
```

# Primjer: škola (if – elif – else – while) (1)

- ▶ Napiši program koji će za unesenu starost osobe reći ide li u osnovnu ili srednju školu ili na fakultet. Program mora tražiti unos godina **sve dok se ne unese negativna vrijednost.**

```
for i in range(5):  
    dijete_staro = input('Unesi starost osobe: ')  
    i = 0
```

```
    if dijete_staro < 0:  
        print('Unesena negativna vrijednost. Program završava.')  
        break  
    elif dijete_staro >= 7 and dijete_staro <= 14:  
        print('Osoba ide u osnovnu školu.')  
    elif dijete_staro >= 15 and dijete_staro <= 18:  
        print('Osoba ide u srednju školu.')  
    elif dijete_staro >= 19 and dijete_staro <= 23:  
        print('Osoba ide na fakultet.')  
    else:  
        print('Osoba ne ide ni u školu, ni na fakultet.')  
    i += 1
```

```
dijete_staro = 0
```

```
while dijete_staro >= 0:  
    dijete_staro = input('Unesi starost osobe: ')  
    if dijete_staro >= 7 and dijete_staro <= 14:  
        print('Osoba ide u osnovnu školu.')  
    elif dijete_staro >= 15 and dijete_staro <= 18:  
        print('Osoba ide u srednju školu.')  
    elif dijete_staro >= 19 and dijete_staro <= 23:  
        print('Osoba ide na fakultet.')  
    else:  
        print('Osoba ne ide ni u školu, ni na fakultet.')  
    dijete_staro = 0
```

# Primjer: škola (if – elif – else – while) (2)

---

```
dijete_staro = 0

while dijete_staro >= 0:
    dijete_staro = input('Unesi starost osobe: ')

    if dijete_staro >= 7 and dijete_staro <= 14:
        print('Osoba ide u osnovnu skolu.')
    elif dijete_staro >= 15 and dijete_staro <= 18:
        print('Osoba ide u srednju skolu.')
    elif dijete_staro >= 19 and dijete_staro <= 23:
        print('Osoba ide na fakultet.')
    else:
        print('Osoba ne ide ni u skolu, ni na fakultet.')
```

```
>>>
Unesi starost osobe: 12
Osoba ide u osnovnu skolu.
Unesi starost osobe: 22
Osoba ide na fakultet.
Unesi starost osobe: -2
Osoba ne ide ni u skolu, ni na fakultet.
>>>
```

# Primjer: škola (if – elif – else – while) (3)

---

- ▶ Neka se nakon upisa negativnih godina, program odmah prekine.

```
dijete_staro = input('Unesi starost osobe: ')
while dijete_staro >= 0:
    if dijete_staro >= 7 and dijete_staro <= 14:
        print('Osoba ide u osnovnu skolu.')
    elif dijete_staro >= 15 and dijete_staro <= 18:
        print('Osoba ide u srednju skolu.')
    elif dijete_staro >= 19 and dijete_staro <= 23:
        print('Osoba ide na fakultet.')
    else:
        print('Osoba ne ide ni u skolu, ni na fakultet.')
    dijete_staro = input('Unesi starost osobe: ')
```

```
>>>
Unesi starost osobe: 12
Osoba ide u osnovnu skolu.
Unesi starost osobe: 22
Osoba ide na fakultet.
Unesi starost osobe: -5
>>>
```