

PROGRAMIRANJE 1

VJEŽBA 3: IF-ELIF-ELSE

SADRŽAJ

UVJETNA NAREDBA: IF-ELIF-ELSE

Jednostruko grananje (IF)

Dvostruko grananje (IF-ELSE)

Ugnježđivanje (If-ELIF)

Višestruko grananje (IF-ELIF-ELSE)

ZADACI

IF-ELIF-ELSE

- Ukoliko se u programu određeni slijed naredbi mora izvršiti samo ako je zadovoljen određeni uvjet, koristimo naredbe **grananja** i razlikujemo jednostruko, dvostruko ili višestruko grananje.
- **Jednostruko grananje (IF)**

if uvjet: → Iza uvjeta obavezno slijedi dvotočka!

blok_naredbi → Blok naredbi koji se izvršava u slučaju ispunjavanja uvjeta mora biti uvučen (tipkom TAB)

└

Python koristi uvlačenje kao metodu razlikovanja programskih blokova! Prethodni *blok_naredbi* se izvršava ako i samo ako je ispunjen *uvjet* (ako rezultira sa True (1)), dok se inače “preskače”.

IF-ELIF-ELSE

- Dvostruko grananje (IF-ELSE)

if *uvjet*:

blok_naredbi

else:

blok_naredbi2

U slučaju da *uvjet* nije zadovoljen te rezultira sa vrijednosti False (0), izvršava se *blok_naredbi2*.

Ne zaboravite dvotočku!

- Prethodna if-else struktura se u svakom slučaju izvršava, bilo to if grananje ili else grananje.

IF-ELIF-ELSE



```
#PrviPrimjer.py
```

```
a=3
```

```
b=86
```

```
if a>b:
```

```
    print('{} je vece od {}'.format(a,b))
```

```
#DrugiPrimjer.py
```

```
a=3
```

```
if a%2==0:
```

```
    print('{} je paran broj.'.format(a))
```

```
else:
```

```
    print(a,'je neparan broj.')
```

IF-ELIF-ELSE

- Ugnježđivanje (IF-ELIF)

Za razliku od IF-ELSE dvostrukog grananja u kojem se prilikom izvršavanja programa uvijek ispuni jedno od dva grananja (jedan od uvjeta), moguće je programirati i ugnježđivanje u kojem se provjeravaju 2 uvjeta, ali se nijedno grananje ne mora i izvršiti.

if *uvjet*:

blok_naredbi

elif *uvjet2*:

blok_naredbi2

IF-ELIF-ELSE

- Višestruko grananje (IF-ELIF-ELSE, IF-ELIF)

if *uvjet*:

blok_naredbi

elif *uvjet2*:

blok_naredbi2

elif ...

else:

blok_naredbiN

Python izvršava liniju po liniju programskog koda. Provjerava se prvi *uvjet* – ukoliko je istinit, izvršava se *blok_naredbi* i ostatak grananja se dalje ne provjerava niti izvršava; ukoliko prvi *uvjet* nije istinit, provjerava se sljedeći *uvjet2*, itd.

Nije nužno da u ovakvoj strukturi postoji ELSE grana – može biti i višestruko IF-ELIF grananje.

IF-ELIF-ELSE



```
#TreciPrimjer.py
```

```
a=int(input("Unesi cijeli broj:"))
```

```
if a>0:
```

```
    print(a,'je pozitivan broj')
```

```
elif a==0:
```

```
    print(a,'je jednak nuli.')
```

```
else:
```

```
    print(a,'je negativan broj.')
```



Kada unutar uvjetnih naredbi provjeravamo logički uvjet jednakosti, obavezno je pisanje dvostrukog znaka jednakosti (==)!

Jedan znak jednakosti (=) Python prepoznaje kao pridruživanje vrijednosti.

ZADATAK 3.1.

- Napišite program pomoću kojeg ćete automatski ispisivati pozdrav prilikom pisanja neke nove poruke. Korisnik unosi trenutno vrijeme (npr. 15.25, 10.00, 23.15) i poruku, a program ispisuje pozdrav i poruku:
 - od ponoć do 9.59 sati ispisuje se pozdrav “Dobro jutro”
 - od 10 do 19.59 sati ispisuje se pozdrav “Dobar dan”
 - a inače se ispisuje pozdrav “Dobra večer”.
- Što ukoliko se za sat unesu sljedeći podaci:
-3, 25, “sat”?

ZADATAK 3.1.

- Ispis:

```
>>>
Unesite sat u obliku sati.minuta:6.50
Unesite poruku:Molim Vas informaciju gdje se održava sutrašnji sastanak? Hvala, Maja
Dobro jutro,
Molim Vas informaciju gdje se održava sutrašnji sastanak? Hvala, Maja
>>> ===== RESTART =====
>>>
Unesite sat u obliku sati.minuta:14.30
Unesite poruku:Molim Vas informaciju gdje se održava sutrašnji sastanak? Hvala, Maja
Dobar dan,
Molim Vas informaciju gdje se održava sutrašnji sastanak? Hvala, Maja
>>> ===== RESTART =====
>>>
Unesite sat u obliku sati.minuta:22.15
Unesite poruku:Molim Vas informaciju gdje se održava sutrašnji sastanak? Hvala, Maja
Dobra večer,
Molim Vas informaciju gdje se održava sutrašnji sastanak? Hvala, Maja
>>> ===== RESTART =====
>>>
Unesite sat u obliku sati.minuta:25
Unesite poruku:Molim Vas informaciju gdje se održava sutrašnji sastanak? Hvala, Maja
Pogrešno ste unijeli sat.
>>> |
```

- Ukoliko je sat pogrešno unesen, obavijestit ćemo korisnika (ELSE grananje).

ZADATAK 3.2.

- Napišite program koji će za dva unesena broja i operaciju (+, -, * ili /) izračunati njihov zbroj, razliku, umnožak ili kvocijent. Ukoliko je drugi broj nula, ne računati kvocijent, nego ispisati poruku „Dijeljenje s nulom“. Ukoliko je operacija pogrešno unesena, ispisati poruku „Niste dobro unijeli operaciju!“.

ZADATAK 3.2.

- Ispis:

```
>>>
Unesi dva broja:2,3
Unesi zeljenu operaciju:+
Zbroj je: 5
>>> ===== RESTART =====
>>>
Unesi dva broja:2,3
Unesi zeljenu operaciju:-
Razlika je: -1
>>> ===== RESTART =====
>>>
Unesi dva broja:2,3
Unesi zeljenu operaciju:*
Umnozak je: 6
>>> ===== RESTART =====
>>>
Unesi dva broja:2,3
Unesi zeljenu operaciju:/
Kvocijent je: 0.6666666666666666
>>> ===== RESTART =====
>>>
Unesi dva broja:2,0
Unesi zeljenu operaciju:/
Dijeljenje s nulom.
>>> ===== RESTART =====
>>>
Unesi dva broja:5,5
Unesi zeljenu operaciju:z
Niste dobro unijeli operaciju.
>>> |
```

ZADATAK 3.3.

- Napišite program za testiranje prioriteta operatora u matematičkim izrazima.

Program ispisuje poruku “Izračunajte izraz:” i izraz:

$10\%3 \neq (11//4) + 13 \geq \text{True or False}$

- Ukoliko je korisnik pogodio rješenje, ispisati “Pogodak!”
 - ukoliko je pogriješio ispisati “Pogrešan unos!” i dodatnu poruku ovisno o tome je li pogodio ili pogriješio tip podatka (brojčani ili logički tip).
- Koje je točno rješenje izraza? Hoćemo li korisnikov unos spremati kao broj ili kao string?

ZADATAK 3.3.

- Ispis:

```
>>>
Izracunajte izraz:
10 % 3 != (11 // 4) + 13 >= True or False
Unesite rezultat:34
Pogresan unos i tip podatka
>>> ===== RESTART =====
>>>
Izracunajte izraz:
10 % 3 != (11 // 4) + 13 >= True or False
Unesite rezultat:False
Pogresan unos, ali tocan tip podatka!
>>> ===== RESTART =====
>>>
Izracunajte izraz:
10 % 3 != (11 // 4) + 13 >= True or False
Unesite rezultat:True
Pogodak!
>>> ===== RESTART =====
>>>
Izracunajte izraz:
10 % 3 != (11 // 4) + 13 >= True or False
Unesite rezultat:false
Pogodak!
>>> |
```

- Točno rješenje izraza je logička vrijednost True i stoga ćemo korisnikov unos spremati kao string.

ZADATAK 3.4.

- Napišite program koji će od korisnika zatražiti unos tri stringa. Provjeriti jesu li stringovi uneseni prema abecednom redoslijedu, i to uzlazno (A-Z). Također, provjeriti postoje li jednaki među unesenim stringovima.

ZADATAK 3.4.

- Ispis:

```
>>>
Unesite prvi string:zec
Unesite drugi string:pas
Unesite treci string:macka
Stringovi nisu uneseni po abecednom redoslijedu.
Nema jednakih stringova medu unesenima.
>>> ===== RESTART =====
>>>
Unesite prvi string:macka
Unesite drugi string:pas
Unesite treci string:zec
Stringovi su uneseni po abecednom redoslijedu.
Nema jednakih stringova medu unesenima.
>>> ===== RESTART =====
>>>
Unesite prvi string:pas
Unesite drugi string:zec
Unesite treci string:pas
Stringovi nisu uneseni po abecednom redoslijedu.
Postoje dva jednaka stringa.
>>> ===== RESTART =====
>>>
Unesite prvi string:macka
Unesite drugi string:macka
Unesite treci string:macka
Stringovi su uneseni po abecednom redoslijedu.
Unesena su tri jednaka stringa.
>>>
```


ZADATAK 3.5.

- Napišite program koji će za unesenu godinu računati je li godina prijestupna.

Prijestupne godine su sve godine djeljive s 4 i ako nisu djeljive sa 100, kojima se dodaju one koju su djeljive sa 400.

ZADATAK 3.5.

- Ispis:

```
>>>
Unesi godinu za provjeru je li prijestupna:1800
Unesena godina 1800 nije prijestupna!
>>> ===== RESTART =====
>>>
Unesi godinu za provjeru je li prijestupna:2100
Unesena godina 2100 nije prijestupna!
>>> ===== RESTART =====
>>>
Unesi godinu za provjeru je li prijestupna:1600
Unesena godina 1600 je prijestupna!
>>> ===== RESTART =====
>>>
Unesi godinu za provjeru je li prijestupna:2400
Unesena godina 2400 je prijestupna!
>>> |
```

ZADATAK 3.6.

- S tipkovnice se unose 3 jednoznamenkasta broja A, B i C. Potrebno je između operatora '+' i '*' odabrati operator koji se može primijeniti na unesene brojeve kako bi se dobio ispravan matematički izraz.
Na izlazu se ispisuje poruka '+=' ili '*=' ili 'Hokus Pokus' u slučaju da izraz nije moguće složiti.

Ograničenje ulaza: A, B i C su tri jednoznamenkasta broja (0, pozitivni i negativni brojevi) koji se unose s tipkovnice.
- U ovom zadatku pokazano je kako ulazne vrijednosti mogu uključivati određena ograničenja, koja se po unosu vrijednosti odmah moraju provjeriti!

ZADATAK 3.6.

- Ispis:

```
>>>
Unesi 3 broja:4,7,1
Hokus Pokus
>>> ===== RESTART =====
>>>
Unesi 3 broja:2,3,5
+=
>>> ===== RESTART =====
>>>
Unesi 3 broja:4,5,20
A, B i C nisu jednoznamenasti brojevi (0, pozitivni ili negativni)!
>>> ===== RESTART =====
>>>
Unesi 3 broja:3,3,9
*=
>>>
```

ZADATAK 3.7.

- Napišite program koji će za unesene bodove na testu ispisati konačnu ocjenu:

[0-50> Nedovoljan

[50-63> Dovoljan

[63-75> Dobar

[75-88> Vrlo dobar

[88-100] Izvrstan

Ukoliko su bodovi pogrešno uneseni ispisati poruku.

ZADATAK 3.7.

- Ispis:

```
>>>
Unesi broj bodova na testu:90
Izvrstan.
>>> ===== RESTART =====
>>>
Unesi broj bodova na testu:82
Vrlo dobar.
>>> ===== RESTART =====
>>>
Unesi broj bodova na testu:65
Dobar.
>>> ===== RESTART =====
>>>
Unesi broj bodova na testu:50
Dovoljan.
>>> ===== RESTART =====
>>>
Unesi broj bodova na testu:13
Nedovoljan.
>>> ===== RESTART =====
>>>
Unesi broj bodova na testu:134
Pogrešno uneseni bodovi.
>>> |
```

ZADATAK 3.8.

- Napišite program u kojem korisnik unosi radijus kruga. Ukoliko je radijus manji od nula ispisati poruku „Krug ne postoji.“, ukoliko je radijus jednak nula ispisati „Krug je točka.“, a inače izračunati i ispisati površinu kruga zaokruženu na 3 decimale.

ZADATAK 3.8.

- Ispis:

```
>>>
Unesi radijus kruga:-5
Krug ne postoji.
>>> ===== RESTART =====
>>>
Unesi radijus kruga:0
Krug je točka.
>>> ===== RESTART =====
>>>
Unesi radijus kruga:6
Povrsina kruga je: 113.04
>>>
```


ZADATAK 3.9.

- Napišite program koji će učitati dvije točke u trodimenzionalnom prostoru $A=(x_1,y_1,z_1)$ i $B=(x_2,y_2,z_2)$. Ispišite koliki je kvadrat udaljenosti između točaka A i B.

Izračun kvadrata udaljenosti:

$$d^2=(x_2-x_1)^2+(y_2-y_1)^2+(z_2-z_1)^2$$

ZADATAK 3.9.

- Ispis:

```
>>>
Unesi koordinate prve tocke:5,3,3
Unesi koordinate druge tocke:1,1,2
Kvadrat udaljenosti je: 21
>>> |
```

ZADATAK 3.10.

- Napišite program u kojem korisnik unosi realne brojeve a i v .
Na ekranu ispisati volumene i oplošja:

a) valjka radijusa a i visine v ,

$$\text{Baza} = r^2 * 3.14$$

$$\text{Plašt} = 2 * r * 3.14 * v$$

b) pravilne šesterostrane prizme baze a i visine v ,

$$\text{Baza} = 6 *$$

$$\text{Plašt} = 6 * a * v$$

c) pravilne četverostrane prizme baze a i visine v .

$$\text{Baza} = a^2$$

$$\text{Plašt} = 4 * a * v$$

$$\text{Opseg} = 2 * \text{Baza} + \text{Plašt}$$

$$\text{Volumen} = \text{Baza} * v$$

ZADATAK 3.10.

- Ispis:

```
>>>
Unesi realni broj a:2.3
Unesi realni broj b:1
Oplosje valjka je: 47.6652 Volumen valjka je: 16.610599999999998
Oplosje 6-strane prizme je: 41.255099999999999 Volumen 6-strane prizme je: 13.727549999999997
Oplosje 4-strane prizme je: 19.779999999999998 Volumen 4-strane prizme je: 5.289999999999999
>>>
```