

Igor Buzov, dipl. inf., viši predavač
Centar umjetne inteligencije Lipik

If - else izrazi



AI Center Lipik
THE FUTURE IS HERE

Pregled lekcije

- U ovoj lekciji ćemo se upoznati s if – else petljama i kako na osnovu njih možemo granati naš program u nekoliko smjerova
- Pokazat će se kada se koristi if – else izraz, a kada if – elif – else izraz
- Pokazat ćemo kako se kombiniraju if – else izrazi s operatorima in i not in
- If izrazi se mogu pojaviti i u ugnježđenom obliku

Ponavljanje – što će biti rezultat ovog koda

```
predmeti = ["olovka", "papir", "maramica", "spajalica", "gumica", "uteg"]
```

```
print (predmeti[1], predmeti[3])
```

```
print (predmeti[-2])
```

```
print (predmeti[1:3])
```

Ponavljjanje – što će biti rezultat ovog koda

```
rijeke = ["Sava", "Drava", "Kupa", "Krka"]  
print (rijeke)
```

```
rijeke.append("Dunav")  
print (rijeke)
```

```
rijeke.insert (1, "Dobra")  
print (rijeke)
```

```
rijeke = ["Sava", "Drava", "Kupa",  
"Krka"]
```

```
rijeke.remove("Kupa")  
print (rijeke)
```

```
rijeke.pop(1)  
print (rijeke)
```

Ponavljanje – što
će biti rezultat
ovog koda?

```
def upis_studenta(ime):  
    studenti.append(ime)  
    studenti.sort()
```

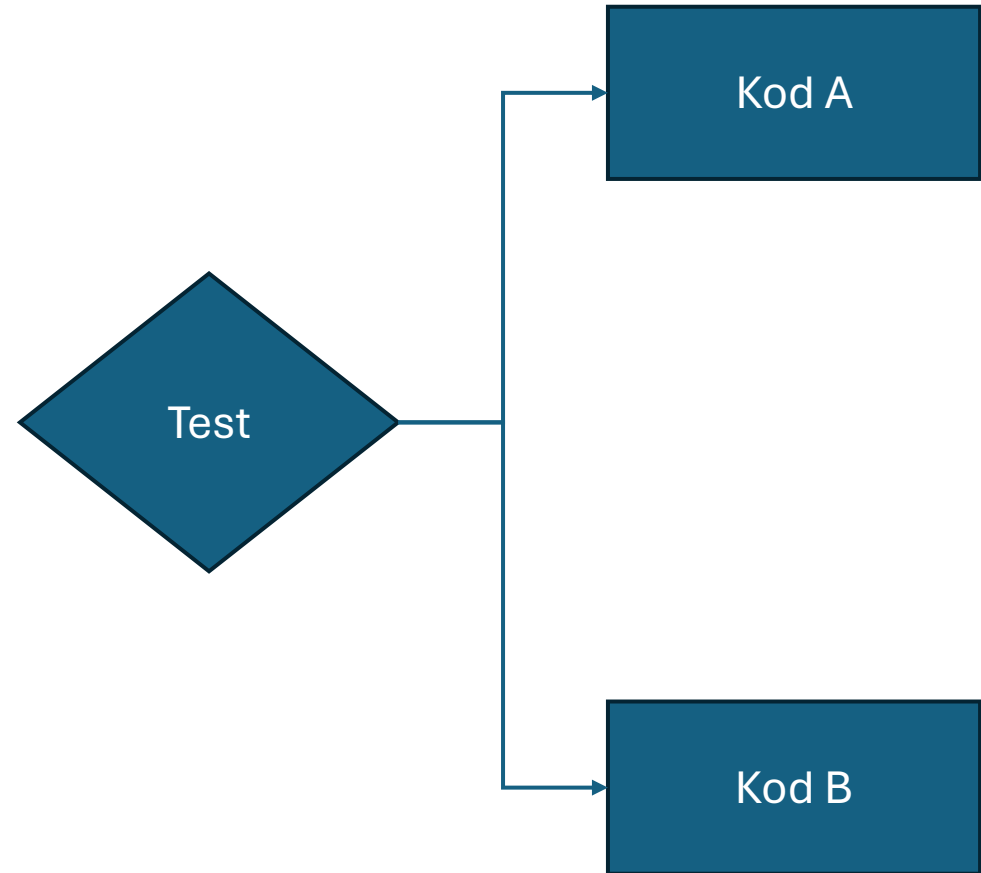
```
studenti = ["Anić Ana", "Perić Pero", "Tomić  
Tomo"]  
print("Trenutačno upisani studenti:")  
print(studenti)
```

```
novi_student = input("Unesite prezime i ime  
novog studenta: ")
```

```
upis_studenta(novi_student)  
print(studenti)
```

if – else izrazi u Pythonu

- If – else odluke (izrazi) nam omogućuju donošenje odluka u Pythonu, na temelju određenih uvjeta
- Izvršavaju se različiti dijelovi koda ovisno o postavljenim pravilima (testovima)



Izrada novog file-a za vježbanje

Otvorite novi file na nazovite ga ifelseizrazi.py

U njemu ćemo vježbati primjere i zadatke ove lekcije

Osnovni if – else izraz

```
broj1 = 37
```

```
broj2 = 10
```

```
if broj1 > broj2:
```

```
    print ("Prvi broj je veći!")
```

```
else:
```

```
    print ("Drugi broj je veći!")
```

- Na početku smo definirali neke dvije varijable, želimo vidjeti koja vrijednost je veća
- Počinjemo s if testom (nekakva usporedba)
 - Pazi na dvotočku i uvlačenje
- Ako je prvi izraz točan, ispisuje se tekst "Prvi broj je veći!". Drugi dio (else) se neće izvršiti
- Ako prvi izraz nije točan, preskače se prvi dio koda i skače se else dio
- REZULTAT: ispisati će se:
 - "Prvi broj je veći!"

Primjer 1 – korištenje jednostavnog if – else izraza

pripazi na uvlačenje koda kao i na dvotočke

```
auto = input("Unesite naziv vaše  
omiljene marke automobila:  
").lower()
```

```
if auto == "bmw":  
    print("Volite prave njemačke aute")  
else:  
    print("Niste ljubitelj njemačkih  
auta!")
```

- Objašnjenje koda – pitamo korisnika za njegovu omiljenu marku vozila
- Koristimo lower() metodu jer ne znamo na koji način će korisnik napisati odgovor (Python je case sensitiv)
- Pazi, koristimo ==, a ne =
- Ovisno o njegovom odgovoru, ispisuje se tekst o tome voli li korisnik prave njemačke aute



Operator "in"

- Što ako naši korisnici iz prijašnjeg primjera navedu Audi ili Mercedes? To su isto pravi njemački auti
 - Koristimo operator "in"
 - Operator in se koristi za provjeru prisutnosti nekog elementa unutar sekvence, kao što su liste, nizovi (stringovi), torke (tuple) ili skupovi (set)
 - Vraća True ako je element prisutan u sekvenci, a False ako nije.
-

Primjer 2

```
omiljeni_auti = ["bmw", "audi", "mercedes"]
```

```
auto = input("Unesite naziv vaše omiljene marke automobila: ").lower()
```

```
if auto in omiljeni_auti:
```

```
    print("Volite prave njemačke aute")
```

```
else:
```

```
    print("Niste ljubitelj njemačkih auta!")
```

Zadatak 1

- Kreirajte varijablu pod nazivom broj1 i pohranite neki cijeli broj u nju
- Zatražite od korisnika da unese neki broj i njegov unos pohranite u varijablu broj2
- Korištenjem if – else izraza ispišite na ekranu koji broj je veći
 - Ako je broj1 \geq broj 2, ispisuje se tekst: "Moj broj je veći!"
 - Inače se ispisuje tekst: "Tvoj broj je veći!"

Zadatak 2

- Šifra za naš wifi glasi: Lipik_online123
- Zatražite od korisnika da unese šifru i njegov odgovor pohranite u odgovarajuću varijablu
- Ako njegova šifra odgovara našoj šifri za wifi, ispišite tekst: vaša šifra je ispravna
- Inače se ispisuje tekst: Neispravna šifra, probajte ponovo!

Zadatak 3

- Napravite listu pod nazivom `dostupni_proizvodi` i popunite je s nekoliko naziva voća (banane, jabuke, kruške...)
- Zatražite od korisnika da unese koji proizvod želi kupiti i pohranite vrijednost u odgovarajuću varijablu
- Ako se proizvod koji je unio korisnik nalazi na popisu dostupnih proizvoda, ispisati: "Proizvod je dostupan"
- Inače ispisati: "Žao nam je, trenutno proizvod nije u ponudi!"

Korištenje operatora not in

- U testiranju nalazi li se neki element u listi, možemo koristiti i operator "not in"
- "not in" testira nedostupnost nekog elementa u listi
- Vrlo često jedan te isti problem možemo riješiti i korištenjem operatora "in" i korištenjem operatora "not in"

Primjer 3

Kod s korištenjem operatora "in"

```
omiljeni_auti = ["bmw", "audi", "mercedes"]
```

```
auto = input ("Unesite naziv vaše omiljene marke  
automobila: ").lower()
```

```
if auto in omiljeni_auti:
```

```
    print ("Volite prave njemačke aute")
```

```
else:
```

```
    print ("Niste ljubitelj njemačkih auta!")
```

Kod s korištenjem operatora "not in"

```
omiljeni_auti = ["bmw", "audi", "mercedes"]
```

```
auto = input ("Unesite naziv vaše omiljene marke  
automobila: ").lower()
```

```
if auto not in omiljeni_auti:
```

```
    print ("Niste ljubitelji njemačkih auta")
```

```
else:
```

```
    print ("Volite prave njemačke aute")
```


Zadatak 4

- Napravite listu s popisom baniranih korisnika foruma
 - Zatražite od korisnika da unese svoj login i pohranite odgovor u odgovarajuću varijablu
 - Provjerite da se dobiveni login NE nalazi u listi baniranih korisnika foruma
 - Ako se logih NE nalazi na listi, ispisuje se tekst:
 - "Dobro došli na naš forum!"
 - Inače se ispisuje tekst:
 - "Zabranjen vam je pristup našem forumu!"
-

Korištenje višestrukih uvjeta

- Postoje situacije gdje u if dijelu želimo testirati više uvjeta
 - Koristimo "and" i "or" operatore
 - and operator – svi uvjeti moraju biti zadovoljeni
 - or operator – dovoljno je da jedan operator bude zadovoljen
-
- And i or operatori su rezervirane riječi, obavezno se pišu malim slovima
 - Možemo koristiti zagrade radi veće čitljivosti, ali to nije obvezno, to je stvar osobnog stila

Primjer 4 – korištenje višestrukih uvjeta

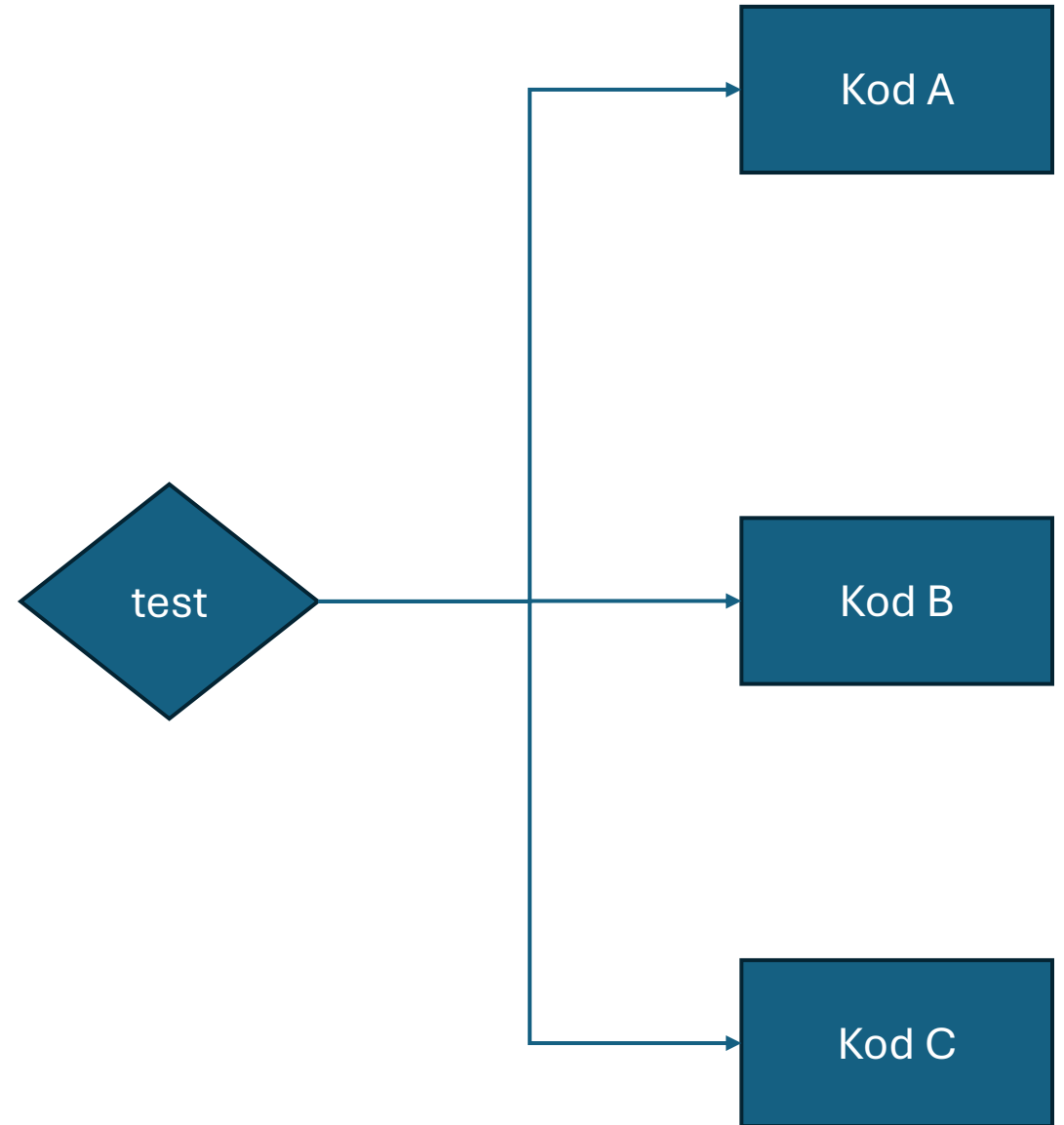
```
godine = int (input ("Koliko imate godina? "))  
vozacka = input ("Imate li vozačku dozvolu? da/ne").lower()  
  
if (godine >= 18) and (vozacka == "da"):  
    print ("Možete voziti osobni automobil!")  
else:  
    print ("Nije vam dozvoljeno voziti automobil!")
```

Zadatak 5

- Imate igraonicu za djecu od 8 do 12 godina
- Pitajte korisnika za starost djeteta i pohranite odgovor u odgovarajuću varijablu
- Postavite uvjet da starost djeteta mora biti između 8 i 12 godina (koristite odgovarajući operator)
- Ako dijete zadovoljava uvjet, ispišite tekst:
 - "Dobro došli, možete se igrati!"
- Inače ispišite tekst:
 - Nažalost ne ispunjavate uvjete, ne možete pristupiti!"

If – elif – else izraz

- Vrlo često želimo testirati više od dvije moguće situacije i tada koristimo sintaksu if – elif – else
- U realnom svijetu vrlo često imamo više od dvije opcije
- Izvršavaju se testovi, jedan za drugim dok se ne pronade ona koji zadovoljava uvjet (makar to bila zadnja) i tada se staje s testiranjem uvjeta



Primjer 5 – if – elif – else izraz (cijena karte)

```
starost = int(input("Koliko imate godina? "))
```

```
if starost < 7:
```

```
    print("Cijena ulaznice iznosi 0 eura")
```

```
elif starost < 18:
```

```
    print("Cijena ulaznice iznosi 10 eura")
```

```
else:
```

```
    print("Cijena ulaznice iznosi 20 eura")
```

- Unost starosti posjetitelja
- Testira se prvi uvjet, je li osoba mlađa od 7 godina
 - Ako je taj uvjet istinit, ulaz za posjetitelja je besplatan
- Ako prvi uvjet NIJE zadovoljen, prelazi se na sljedeći dio koda
- Python je naučio da je osoba starija od 7 godina, testira je li mlađa od 18 godina
- Ako ni drugi test nije zadovoljen, tek onda se ide na treći dio koda

Primjer 6 – Pythonic way of solving the problem

```
starost = int (input ("Koliko imate godina? "))

if starost < 7:
    cijena = 0
elif starost < 18:
    cijena = 10
else:
    cijena = 20

print ("Cijena ulaznice iznosi: ", cijena)
```

Zadatak 6

- Studenti su pisali ispit, na ispitu je moguće ostvariti između 0 i 100 bodova
- Zatražite od korisnika da unese broj bodova na ispitu
- Ako je student ostvario 90 i više bodova, dobit će ocjenu "A"
- Ako je student ostvario 75 i više bodova, ostvarit će ocjenu "B"
- Inače mu se dodjeljuje ocjena "C"

Primjer 7

if – elif – else izraz

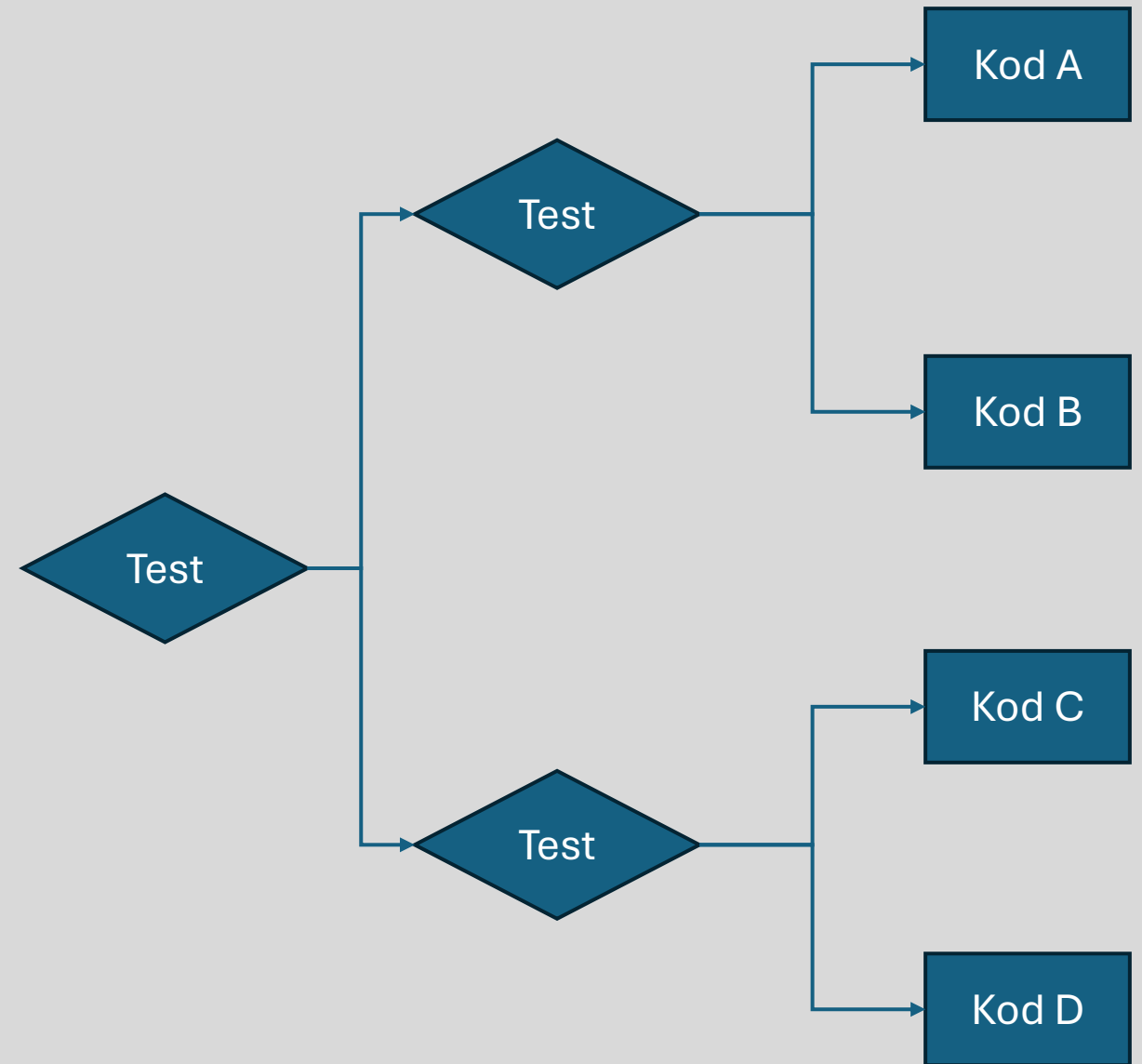
- If – elif – else izraz vam omogućuje i više mogućih elif blokova, koliko god vam treba, ne samo jedan

```
bodovi = int(input("Unesi broj bodova (0-100): "))
```

```
if bodovi >= 90:  
    print("Ocjena: 5")  
elif bodovi >= 75:  
    print("Ocjena: 4")  
elif bodovi >= 60:  
    print("Ocjena: 3")  
elif bodovi >= 50:  
    print("Ocjena: 2")  
else:  
    print("Ocjena: 1")
```

Ugnježdjeni if izrazi

- Ugnježdjeni if izrazi predstavljaju situaciju gdje unutar jednog if bloka postoji još jedan if blok s nekoliko podopcija
- Provjeravamo dodatne uvjete na osnovu prethodno ispunjenih uvjeta



Primjer 8 – ugnježđeni if izrazi

```
narudzba = input("Želite li obrok ili samo piće? (obrok/piće): ").lower()

if narudzba == "obrok":
    hrana = input("Želite li menu ili pojedinačno jelo? (menu/jelo): ").lower()
    if hrana == "menu":
        print("Preporučamo naš mesni menu!")
    else:
        print("Preporučamo hamburger.")
else:
    napitak = input("Želite li kavu ili sok? (kava/sok): ").lower()
    if napitak == "kava":
        print("Preporučamo našu kavu s mlijekom.")
    else:
        print("Preporučamo našu cijedenu limunadu.")
```

Zadatak 7

- Pogledati zadatak u datoteci
 - 07 zadatak filmovi ugnjezdjeni if.pdf

Inkrementiranje varijable - primjer

```
cijena_nocnja = 100
```

```
ljubimac = input("Imate li kućnog ljubimca? da/ne ").lower()
```

```
if ljubimac == "da":
```

```
    cijena_nocnja = cijena_nocnja + 20
```

```
    print ("Cijena noćenja iznosi: ", cijena_nocnja)
```

```
else:
```

```
    print ("Cijena noćenja iznosi: ", cijena_nocnja)
```

+= operator inkrementiranja

```
cijena_nocenja = 100
```

```
ljubimac = input("Imate li kućnog ljubimca?  
da/ne ").lower()
```

```
if ljubimac == "da":
```

```
    cijena_nocenja += 20
```

```
    print ("Cijena noćenja iznosi: ", cijena_nocenja)
```

```
else:
```

```
    print ("Cijena noćenja iznosi: ", cijena_nocenja)
```

Zadatak 8

- Trgovina "Sve za dom" nudi popust na ukupnu cijenu narudžbe ovisno o tome je li kupac član loyalty programa.
- Postavite početnu ukupnu cijenu narudžbe na neki iznos (npr. 500).
- Pitajte korisnika je li član loyalty programa. Omogućite unos "da" ili "ne" (neovisno o velikim i malim slovima).
- Ako je kupac član loyalty programa:
 - Smanjite ukupnu cijenu narudžbe za 50 eura koristeći operator -=.
 - Ispišite poruku koliki je konačni iznos narudžbe.
- Ako kupac nije član loyalty programa:
 - Ispišite poruku da popust nije primijenjen

Korištenje if – else izraza u vlastitim funkcijama

- Modularnost: Kada želite da vaš kod bude modularan i lako održiv, stavljanje if-else logike unutar funkcija pomaže u organizaciji koda.
 - Ponovljivost: Ako imate logiku koja se ponavlja u više dijelova programa, stavljanje te logike u funkciju s if-else čini kod čistijim i lakšim za održavanje.
 - Čitljivost: Funkcije s if-else strukturama čine kod lakšim za razumijevanje jer su odvojene od glavnog toka programa.
-

Korištenje if – else izraza u vlastitim funkcijama – primjer Online bankarstvo

- U online bankarstvu možete raditi različite operacije
 - Konverzija valuta
 - Plaćanje računa
 - Transfer sredstava s jednog računa na drugi (tekući/žiro)
 - Slanje novca drugoj strani
- Unutar svake od tih operacija, provodi se provjera radnog vremena
- Umjesto pisanja jednog te istog koda četiri puta, u svakoj operaciji se može pozvati funkcija za provjeru radnog vremena
- Pogledati primjer u datoteci:
 - 07 primjer Online bankarstvo.pdf

Sažetak lekcije

- If – else izrazi u Pythonu su kontrolne strukture koje omogućuju izvođenje različitog koda ovisno o ispunjenju uvjeta
 - if provjerava uvjet i ako je istinit, izvršava određeni blok koda
 - else omogućuje izvođenje alternativnog koda ako je uvjet neistinit
- if-elif-else struktura u Pythonu omogućuje provjeru više uvjeta.
 - if provjerava prvi uvjet
 - elif dodatne uvjete ako je prvi neistinit
 - else izvršava kod ako nijedan uvjet nije istinit
- Ugnježđeni if izrazi se koriste kada je potrebno provjeriti dodatne uvjete unutar već postojećeg if bloka.