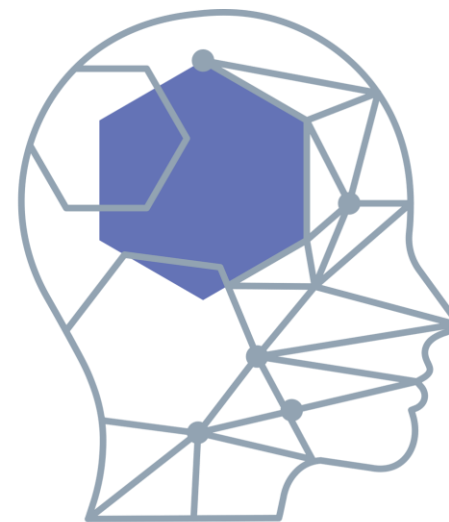


Varijable i jednostavni tipovi podataka (stringovi)

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



AI Center Lipik
THE FUTURE IS HERE

Pregled lekcije

- Upoznati se s varijablama i njihovim tipovima
 - Da, varijable mogu biti različitog tipa 😊
- Upoznati se s metodama i operacijama na metodama



Ponavljanje

- Za programiranje u Pythonu možemo koristiti Word ili Powerpoint? T/N
 - Što je IDE? Čemu služi? Navedite neki od njih ...
 - Želite snimiti file s Python kodom. Kako ćete ga imenovati?
 - a. proba.pyt
 - b. proba.py
 - c. proba.pht
-

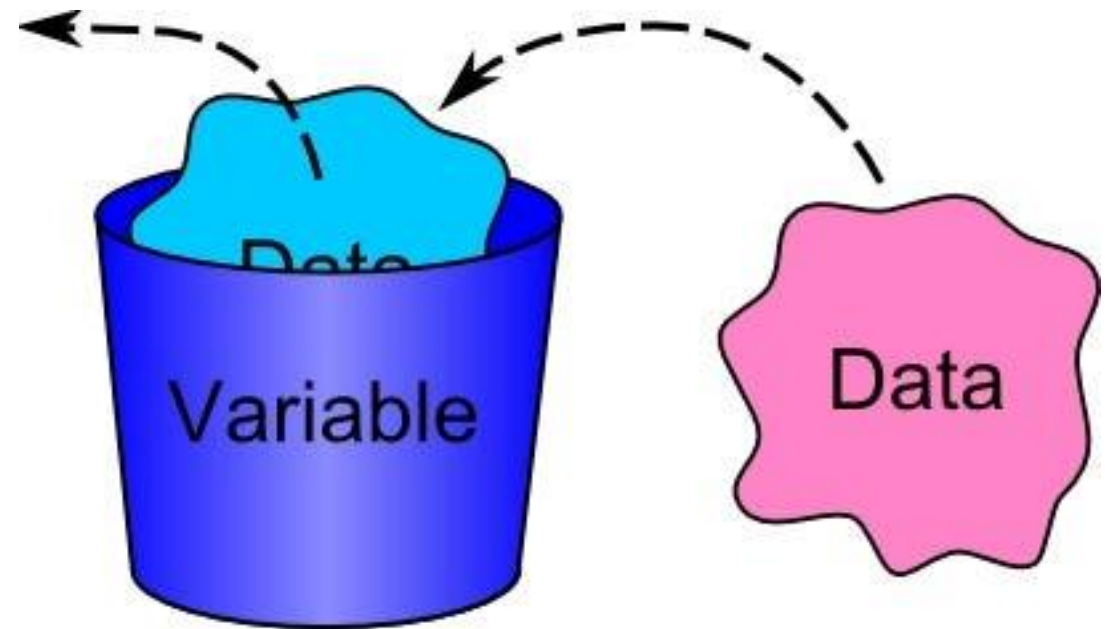
Što su varijable

- Varijabla je simboličko ime koje dajemo nekoj vrijednosti
- Tu vrijednost smo pohranili pod tim imenom u svojevršno skladište
- Primjer koda

ime = "Ana"

broj = 100

- Ime je varijabla koja sadrži tekst "Ana"
- Broj je varijabla koja sadrži broj 100
- Primjetite da smo jedanput oko vrijednosti stavili navodnike, a drugi put nismo (tekst vs broj)



Program hello_world.py

- Modificirajte vaš program te ga pokrenite:

```
message = "Hello Python world!"
```

```
print(message)
```

- Što smo dobili?
 - Pohrana podataka: varijable nam omogućuju da pohranimo podatke koje želimo koristiti u našem programu
 - Ponovno korištenje: umjesto da svaki put pišemo istu vrijednost, možemo koristiti ime varijable
 - Čitljivost koda: varijable čine kod čitljivijim, jer umjesto brojeva ili tekstova imamo značajna imena koja opisuju njihovu svrhu

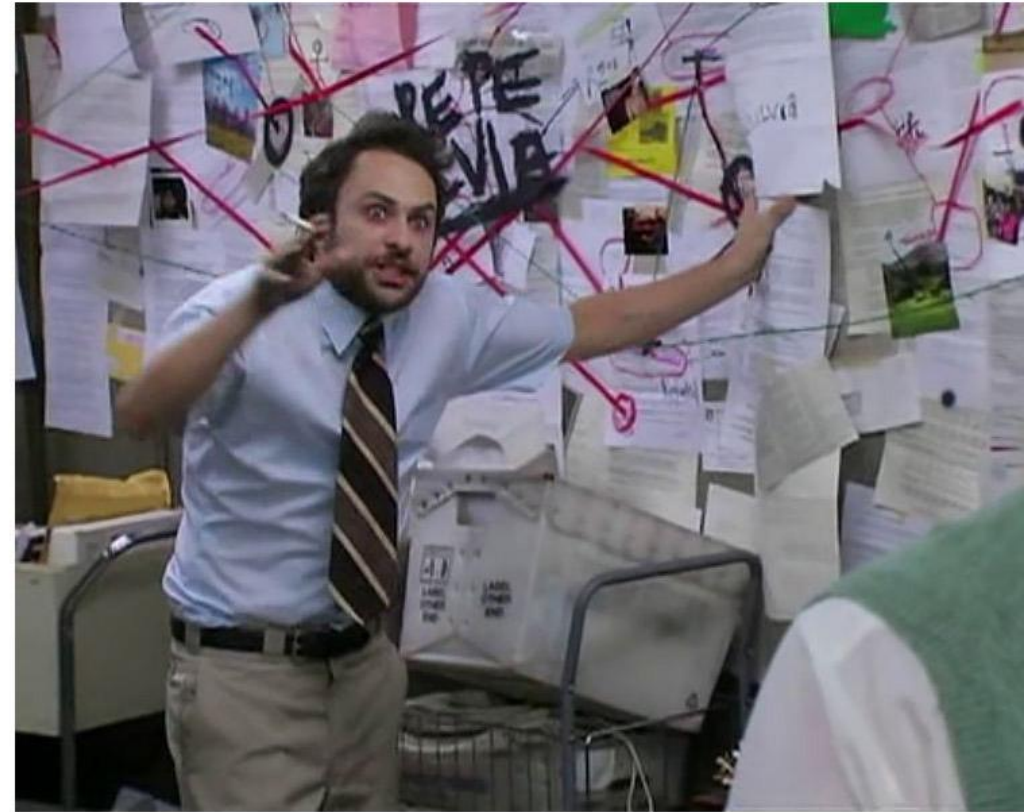
Pravila imenovanja varijabli u Pythonu I

- Slova i brojevi: Imena varijabli mogu sadržavati slova (mala i velika), brojeve i znak donje crte (_).
- Početak imena: Ime varijable mora započeti slovom ili znakom donje crte. Ne može započeti brojem.
- Posebni znakovi: Osim slova, brojeva i donje crte, nisu dopušteni drugi znakovi kao što su razmaci, interpunkcijski znakovi ili matematički operatori.

Pravila imenovanja varijabli u Pythonu II

- Velika i mala slova: Python razlikuje velika i mala slova. Tako varijable broj, Broj i BROJ predstavljaju tri različite varijable.
- Ključne riječi: Ne mogu se koristiti ključne riječi Pythona kao imena varijabli. To su riječi koje imaju posebno značenje u jeziku, poput if, else, for, while, def, itd.
- Značajna imena: Iako Python ne zahtijeva da imena varijabli budu opisna, dobra praksa je koristiti imena koja jasno ukazuju na namjenu varijable.

Me explaining my variable naming scheme to the other devs



Primjeri – odredite dobre i loše nazive varijabli

- broj_studenata
- for
- temperatura
- 1broj
- ime studenta
- ime_prezime
- _privatna_varijabla
- broj%2
- ime_studenta_i_njegovo_prezime
- broj7
- z
- Grad
- sum_rac_zg_K
- županija



TRY IT YOURSELF

Zadatak 1

- Otvorite novi file i napravite u njemu četiri varijable u kojima ćete pohraniti podatke o svom imenu, mjestu rođenja, godinama i visini
- Pazite gdje koristite navodnike (tekst vs broj)
- Ispišite vrijednosti korištenjem naredbe `print ()`
- ***Bonus: Kako biste ispisali vrijednosti svih varijabli korištenjem samo jedne naredbe `print`?***

Program hello_world.py

- Ispod vašeg koda dodat ćete još dvije linije koda

```
message = "Hello Python world!"
```

```
print(message)
```

```
message = "Hello AI Lipik"
```

```
print(message)
```

- Tokom izvođenja programa možete mijenjati vrijednost vaših varijabli

Rad s greškama

- Nekoliko pravila vezana za greške koje se događaju pri programiranju:
 1. Greške se događaju
 2. Greške su normalne
 3. Svima se događaju greške
 4. Pročitaj ponovo prvo pravilo
- Kada dođe do greške, Python se trudi, najbolje što može, ukazati na mjesto gdje se desila greška

hello_world.py 1 X podaci.py

D: > IGOR > Posao > LIPIK AI > LIPIK Python > python_work > hello_world.py > ...

```
1 message = "Hello World"
2 print (mesage)
3
```

PROBLEMS

1

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

```
PS C:\Users\Admin> & C:/Users/Admin/AppData/Local/Programs/Python/Python312/python.exe "d:/IGOR/Posao/LIPIK AI/LIPIK Python/python_work/hello_world.py"
```

```
Traceback (most recent call last):
```

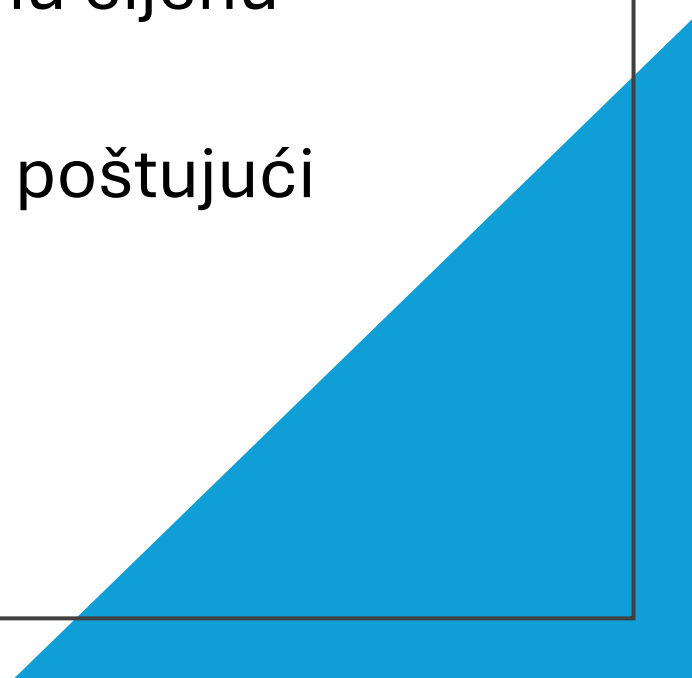
```
File "d:\IGOR\Posao\LIPIK AI\LIPIK Python\python_work\hello_world.py", line 2, in <module>
    print (mesage)
          ^^^^^^
```

```
NameError: name 'mesage' is not defined. Did you mean: 'message'?
```

```
PS C:\Users\Admin>
```

Moguća greška u liniji koda 2, ne postoji varijabla mesage

Zadatak 2 – odredite varijable

- **Iznajmljivanje automobila**
 - Osoba želi iznajmiti automobil na nekoliko dana. Cijena najma po danu je 50 eura. Osoba ima popust od 10% jer je član kluba. Program treba izračunati konačnu cijenu najma.
 - Odredite bitne varijable, imenujte ih smisleno poštujući pravila imenovanja varjabli u Pythonu
 - Proširite scenarij i dodajte nove varijable!
- 
- A large blue right-angled triangle is positioned in the bottom right corner of the slide, pointing towards the top right.

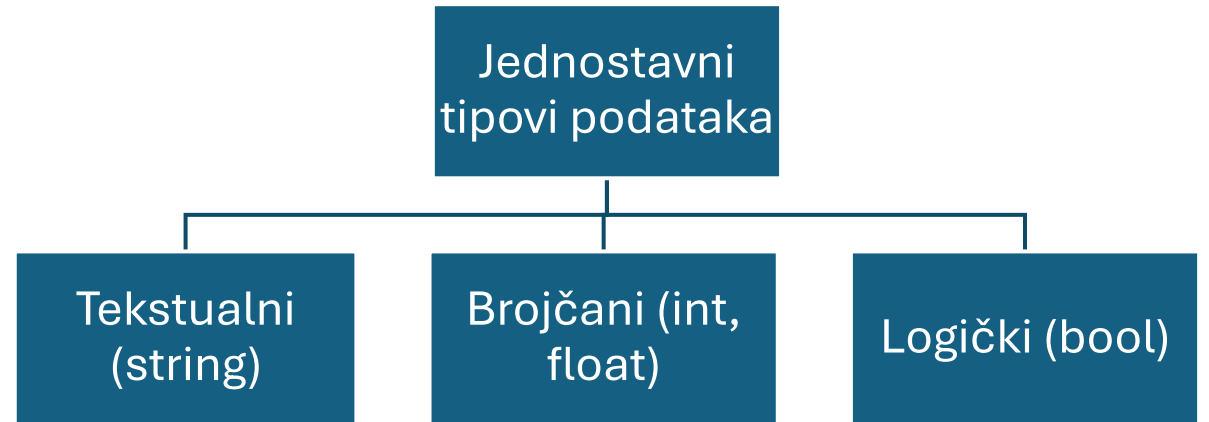
Zadatak 3 – odredite varijable

- **Nogometna utakmica između Dinama i Hajduka**
- Na derbiju između Dinama i Hajduka, Dinamo je postigao 3 gola, dok je Hajduk uspio zabiti 2 gola. Želimo izračunati ukupan broj golova na utakmici te koja je momčad pobijedila.
- Odredite bitne varijable, imenujte ih smisleno poštujući pravila imenovanja varijabli u Pythonu



Jednostavni tipovi podataka

- Programi skupljaju podatke i pokušavaju napraviti nešto korisno s njima
- S brojevima recimo računamo
- Iz teksta pokušavamo izvući značenje
- Zbog toga nam je korisno podatke klasificirati u različite tipove
- Dodjela tipa podatka koji se dodjeljuje se odvija automatski



Tipovi podataka - stringovi

- Prvi tip podatka s kojim ćemo raditi je string
- String je niz znakova. Kad deklariramo varijablu, nakon znaka jednako, sve što se nalazi unutar navodnika se smatra stringom
- Možemo koristiti jednostruke i dvostruke navodnike
- Svaki put kada definiramo varijablu i dodijelimo joj sadržaj koji je omeđen navodnicima, Python će automatski dodijeliti tip string

Ispravan način definiranja varijable koja sadrži string

```
message = "Hello Python world!"
```

```
grad = "Zagreb"
```

```
ulica = "Savska 157"
```

```
zupanija = 'Splitsko – dalmatinska'
```

```
komentar = "Jako mu se svidjelo domaće jelo 'pašticada'! "
```

```
datum = "25.10.2024."
```

Neispravan način definiranja varijable koja sadrži string

```
moj_grad = 'Zagreb"  
ulica = ' 'Varaždinska 15" "  
vladanje = "vrlo dobro
```

Metode na stringovima

- Vrlo često želimo manipulirati tekстом i mijenjati ga
- Primjeri kako je korisnik upisao svoje podatke pri stvaranju accounta
- Igor, IGOR, igor, IGor (za Python su ovo četiri različita korisnika)
- Igor Buzov, Igor Buzov, IgorBuzov, Buzov Igor
- Za manipulaciju stringovima koristimo metode
- Sintaksa:

`Ime_varijable = ime_varijable.ime_metode()`

- Točka i zagrade su obavezni

Metode na stringovima

- Što su zapravo metode?
- Metode su niz uputa Pythonu što da napravi s određenim podacima koje smo mu dali
- Primjeri:
- Možemo Pythonu dati string i reći mu da sva slova postanu mala
- Možemo reći Pythonu da makne razmake prije i poslije neke riječi
- Možemo reći Pythonu da niz riječi u rečenici razdvoji na pojedine riječi
- Zašto su metode korisne?
 - Učinkovitost: metode nam omogućuju da izvršavamo složene operacije na jednostavan način, bez potrebe da sami pišemo dugačke i složene kodove
 - Čitljivost: Korištenje metoda čini kod čitljivijim i lakšim za razumijevanje
 - Modularnost: Metode nam omogućuju razbijanje većih problema na manje, lakše upravljive dijelove

Metoda title() na stringovima

- Metodu title () koristimo kada želimo promijeniti string na način da svaka riječ počinje s velikim slovom, a sva ostala slova (p)ostaju mala
- **Sintaksa:** string.title()
- A ako koristimo varijable: ime_varijable.title()
- Otvorite novi file u VS Codeu, nazovite ga stringovi.py i napišite i pokrenite sljedeći kod

```
glazbenica = "celine dion"  
glazbenica = glazbenica.title()  
print (glazbenica)
```

Metoda title() na stringovima

- U nastavku file – a stringovi.py napišite sljedeći kod:

```
grad = "zagreb"
```

```
drzava = "SJEDINJENE AMERIČKE DRŽAVE"
```

```
ulica = "Iločka ulica 15b"
```

- Korištenjem title() metode, popravite sadržaj varijabli i ispišite ih!
-

Metode upper() i lower () na stringovima

- Ove metode se koriste kako bi sva slova postala velika (upper) ili mala (lower) ovisno o potrebi
- **Sintaksa:**
 - `string.upper()`
 - `string.lower()`
- U vašoj datoteci `stringovi.py` imenujte varijablu `marka_automobila` te primijenite odgovarajuću metodu kako bi sva slova postala velika
- U vašoj datoteci `stringovi.py` imenujte varijablu `povrce` te primijenite odgovarajuću metodu kako bi sva slova postala mala
- Ispišite rezultate

Metode lstrip(), rstrip() i strip()

- Dodatna razmaci (spaces) u vašim programima mogu biti vrlo konfuzni za Python
- "Python" i "Python " su dvije različite varijable
- Vrlo često želite usporediti dva stringa kako biste vidjeli jesu li jednaki, kao što imamo pri logiranju, odgovara li username
- Korištenjem metoda lstrip (micanje spaceova s lijeva), rstrip (micanje spaceova s desna) te strip (micanje spaceova s obe strane) možemo riješiti taj problem

Metode lstrip(), rstrip() i strip()

- **Sintaksa:**

- string.strip()
- string.lstrip
- string.rstrip()

- U vašem programu stringovi.py napravite sljedeće varijable i riješite se razmaka

prvo_mjesto = " Zadar"

drugo_mjesto = "Osijek "

trece_mjesto = " Rijeka "

- Ispišite rezultate!

Metoda split()

- Metoda split() se koristi za razdvajanje stringa na manje podstringove
- Primjeri korištenja:
- Imamo rečenicu teksta (to je isto string) i želimo svaku riječ zasebno izdvojiti (koristimo kod analize teksta)
- Korisnik je unio ime i prezime, želimo podatke razdvojiti u zasebne varijable (varijabla ime i varijabla prezime)





Metoda split()

- **Sintaksa:** `string.split (separator, maxsplit)`
 - Po prvi put se susrećemo s metodom koja ima sadržaj unutar zagrada
 - Sadržaj zagrade se zovu parametri i oni govore metodi kako da obavi svoj zadatak
 - Parametri su ključni koncept u programiranju. Oni omogućuju da funkcije i metode budu dinamične i prilagodljive različitim situacijama. U slučaju metode `split()`, parametri nam omogućuju preciznu kontrolu nad načinom na koji se string razdvaja.
-

Metoda split(separator, maxsplit)

- Parametar separator – s kojim znakom dijelimo na podstringove
 - Ako se izostavi, koristi se razmak
- Parametar maxsplit - maksimalan broj podjela koje će se izvršiti
 - Ako se izostavi, nema ograničenja na broj podjela
 - Kada je maxsplit naveden, broj podstringova će biti navedeni broj + 1
- Po defaultu, rezultat metode split se pohranjuje u listu – o tome što su liste naknadno
 - Mi ćemo rezultate splitanja pohranjivati u varijable

Primjeri

```
ime_i_prezime = "Ivan Horvat"
```

```
ime, prezime = ime_i_prezime.split()
```

- Što će biti sadržaj varijable ime, a što prezime?

```
datum = "2023-11-22"
```

```
godina, mjesec, dan = datum.split("-")
```

- Što će biti sadržaj varijable godina, što mjesec, a što dan?

```
podaci = "Ivana Brlić Mažuranić"
```

```
ime, prezime = podaci.split(" ", 1)
```

- Što će biti sadržaj varijabli ime i prezime?

Zadatak 4 - stringovi.py

- Imenujte varijablu recenica u koju ćete zapisati sljedeći tekst:
"U proljeće, kada cvjetaju voćke, šuma je prepuna pjeva ptica, a livada, prekrivena cvijećem, miriše na med i vaniliju."
- Korištenjem metode split() podijelite ovu rečenicu na dijelove, svaki dio neka se pohrani u posebnu varijablu pod nazivom dio1, dio2, dio3 ...

Zadatak 5

- Kreirajte varijablu po nazivom naselje i u nju pohranite sljedeći tekst:
 - "grad nova gradiška"
 - Kreirajte dvije nove varijable
 - tip_naselja
 - naziv_naselja
 - Korištenjem odgovarajuće metode podijelite sadržaj varijable naselje u dva dijela, tip_naselja i naziv_naselja
 - Korištenjem odgovarajuće metode stavite velika početna slova u varijabli naziv_naselja
 - Ispišite rezultate
-

Ulančavanje metoda

- Radi veće čitljivosti i bolje preglednosti našeg koda, vrlo često koristimo kombiniranje metoda
- Umjesto da primjenimo jednu metodu na stringu pa u novom redu koda drugu, radimo sve u jednom izrazu
- Princip rada je da se nakon prvog izraza stavi nova točka i samo nadoda nova metoda
- Sintaksa: `string.metoda1().metoda2()`

Ulančavanje metoda

```
izraz = "Ovo je primjer rečenice!"  
izraz_velikim slovima = izraz.upper().split()  
print (izraz_velikim slovima)
```

- Prvo se izvodi metoda upper, a nakon nje metoda split
 - Rješenje će biti pohranjeno u objektu koji se zove lista (o tome poslije)
-



Rasprava

- Do koje duljine se preporuča ulančavanje metoda?
- Primjer – je li ovo pregledno? Što ako dodamo još metoda?

```
naslov = naslov.split().strip().lower().title()
```

Konkatenacija – operacija spajanja stringova

- Za potrebe dinamičnog okruženja imamo mogućnost spajanja stringova u jedan niz.
- Koristi se znak +
- U datoteci stringovi.py prepisite sljedeći primjer:

```
ime = "Ivan"  
prezime = "Horvat"  
puno_ime = ime + " " + prezime  
print (puno_ime)
```

- Spajamo naše ime i prezime s još jednim praznim spaceom " " kako se ne bi zalijepili

Zadatak 6 – stringovi.py

- Napravite ove varijable:

`direktorij = "/home/korisnik/dokumenti/"`

`ime_datoteke = "izvjestaj.txt"`

- Napravite novu varijablu pod nazivom putanja koja se sastoji od direktorija i naziva datoteke koja se nalazi u tom direktoriju

Zadatak 7 – stringovi.py

- Napravite varijablu ime_korisnika u koju ćete pohraniti njegovo ime
- Potrebno je maknuti spaceove s obe strane ako postoje!
- Napravite varijablu pozdrav u kojoj je zapisana pozdravna poruka
- Ispišite pozdravnu poruku nakon koje slijedi njegovo ime! Na kraju pozdravne poruke se nalazi uskličnik!

Replikacija – operacija množenja stringova

- Ako želimo više puta ponoviti neki string, koristimo operaciju replikacije
- Koristi se znak *
- U datoteci stringovi.py prepishite sljedeći primjer:

```
sport = nogomet
```

```
pet_puta_nogomet = sport * 5
```

```
print (pet_puta_nogomet)
```

Primjena replikacije u programiranju

- Za neke jednostavne grafičke prikaze, možeš koristiti operator `*` za ponavljanje znakova i stvaranje osnovnih oblika

```
ime = "Ivan"
```

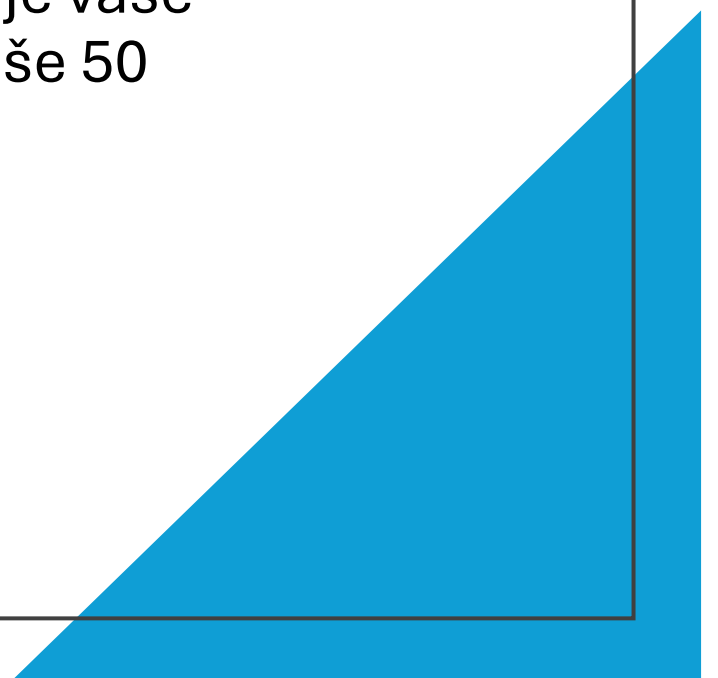
```
poruka = "Pozdrav, " + ime + "." * 3
```

```
print (poruka)
```

- Ovaj operator je posebno koristan kada radiš s tekstualnim podacima i želiš stvoriti ponavljajuće obrasce ili popuniti praznine u tekstu (jedna od metoda za popunjavanje missing values-a)

Zadatak 8 - stringovi.py

- Modificirajte prošli zadatak na način da se prije i poslije vaše pozdravne poruke napiše 50 puta zaredom znak +



1. Zadaća - stringovi

Pronaći u literaturi
vama jednu
zanimljivu metodu na
stringovima te je
objasniti i dati primjer

Rješenje vaše zadaće
napišite u
Powerpointu

Način predaje:

Sažetak lekcije

Varijable nam služe za pohranu podataka pod određenim imenom

Postoje pravila (i preporuke) za imenovanje varijabli

Tekstualni podaci koji se pohranjuju unutar varijabli su tipa string

Za manipulaciju stringovima koristimo metode

Metode: `title()`, `upper()`, `lower()`, `lstrip()`, `rstrip()`, `strip()`, `split()`

Pojedine metode mogu koristiti i parametre (pišu se unutar zagrada)

U jednom izrazu možemo ulančati i više metoda na jednom stringu

Uz metode na stringovima, možemo koristiti i operacije konkatencije (spajanja) i replikacije (množenja)