

Jednostavni tipovi podataka – brojevi i Bool

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



Pregled lekcije



- Uz stringove, u Pythonu imamo i druge tipove podataka kao što su brojevi (cjelobrojni i decimalni) i logički podaci (True i False)
- S brojčanim podacima možemo računati pa ćemo se upoznati s računskim operatorima
- Podaci se mogu i uspoređivati, jesu li jednaki, manji, veći, stoga nam je plan upoznati se s operatorima usporedbe
- U programiranju se preporuča koristiti komentare kako bi kod bio čitljiviji pa ćemo pokazati načine komentiranja koda

Ponavljjanje

Odredite ispravna imena varijabli

mjesto

1981zagreb

_lipik

Mrfe_1458_zgndbegv

mjesto_rodjenja

%kamata%

Lipik

Ispravite greške u kodu

mjesto = 'Zagreb"

print (mjesto

Ponavljjanje - što će biti rezultat ovog koda?

Primjer 1

```
mjesto = "DUBROVNIK"  
mjesto.lower()  
mjesto.title()  
print (mjesto)
```

Primjer 2

```
mjesto = "DUBROVNIK"  
mjesto = mjesto.lower()  
mjesto = mjesto.title()  
print (mjesto)
```

Ponavljjanje –
što će biti
rezultat ovog
koda?

```
osoba = "ivan goran kovačić"  
ime, prezime = osoba.split(" ", 1)  
print (ime)  
print (prezime)  
ime = ime.upper()  
prezime = prezime.title ()  
print (ime)  
print (prezime)
```





Brojevi

- Uz tekstualne podatke, u naše varijable možemo pohranjivati i numeričke s kojima možemo računati
- Za razliku od tekstualnih podataka koji dolaze samo u obliku stringa, numeričkih podataka ima više vrsta
 - integer: cjelobrojni brojevi (npr., 42, -10, 0)
 - float: decimalni brojevi (npr., 3.14, -2.5, 0.0)
 - PAŽNJA: python koristi decimalnu točku

Integers: cijeli brojevi

- Predstavljaju cijele brojeve (pozitivne, negativne i nulu)
- Nemaju decimalni dio
- Koriste se za brojanje, indeksiranje, itd.
- Otvorite novi file u VS Codeu, nazovite ga brojevi.py i napišite sljedeći kod:

```
age = 30
```

```
number_of_apples = 12
```

```
print (age)
```

```
print (number_of_apples)
```

Floats: decimalni brojevi

- Predstavljaju realne brojeve s decimalnim dijelom
- Koriste se za mjerenja, proračune s decimalnim brojevima
- Mogu biti zapisani u decimalnom obliku ili u znanstvenom zapisu
- U fileu brojevi.py dodajte sljedeći kod:

```
pi = 3.14159
```

```
height = 1.75
```

```
print (pi)
```

```
print (height)
```


Paziti na način deklariranja varijabli

- Koja je razlika između deklariranja ove dvije varijable:

`visina = 100`

`duljina = "200"`

- U prvom slučaju radi se o broju, u drugom o stringu
- Ako bismo išli računati površinu ($\text{visina} * \text{duljina}$), Python nam ne bi dao očekivani rezultat

Arifmetičke operacije s numeričkim tipovima (+, -, *, /)

- Kao i u svakom programskom jeziku, Python omogućava operacije s numeričkim tipovima kao što su množenje, dijeljenje, zbrajanje i oduzimanje
- Ažurirajte vaš kod i dodajte sljedeće naredbe:

```
print (age * 2)
```

```
print (number_of_apples / 3)
```

```
print (pi + 5)
```

```
print (height - 100)
```

- Kad ste podijelili varijablu `number_of_apples`, koji je rezultat? Kojeg tipa podatka?

Aritmetičke operacije s numeričkim tipovima (**, //, %)

- Python također podržava i računske operacije potenciranja, cjelobrojnog dijeljenja i modul

```
print(pi ** 3)
```

- Rezultat je konstanta pi potencirana na treću

```
broj_minuta = 632
```

```
broj_sati = broj_minuta // 60
```

- Cjelobrojno dijeljenje koristimo kad nas zanima samo cijeli dio, bez ostatka

```
ostatak = broj_minuta % 60
```

- Modul koristimo kada nas zanima samo decimalni ostatak

Zadatak 1

- Definirajte varijablu brzina_automobila i u nju pohranite neki pozitivni broj
- Definirajte varijablu broj_sati i u nju pohranite neki broj
- Definirajte novu varijablu udaljenost u koju ćete pohraniti koliko pređete automobilom vozeći zadanom brzinom i brojem sati
- Ispišite rezultat

Zadatak 2

- Želimo izračunati površinu kruga.
- Definirajte varijablu promjer i u nju upišite neku vrijednost
- Definirajte varijablu pi i u nju upišite 3.14
- Za izračun površine kruga treba vam polumjer koji dobijete dijeljenjem promjera s brojem 2
- Formula za površinu kruga glasi:
$$\text{povrsina} = \text{polumjer}^2 * \text{pi}$$
- Izračunajte površinu i ispišite rezultat

Operatori usporedbe (>, >=, <, <=, ==, !=)

- Operatori usporedbe nam omogućuju usporedbu dvije vrijednosti, je li jedna veća, manja ili jednaka
- Na osnovi tih usporedbi u kompliciranijim programima izvršavamo određene radnje
- == (jednako): Provjerava je li lijeva strana jednaka desnoj
- != (nije jednako): Provjerava je li lijeva strana različita od desne
- < (manje): Provjerava je li lijeva strana manja od desne
- > (veće): Provjerava je li lijeva strana veća od desne
- <= (manje ili jednako): Provjerava je li lijeva strana manja ili jednaka desnoj
- >= (veće ili jednako): Provjerava je li lijeva strana veća ili jednaka desnoj

Razlika u korištenju između operatora = i ==

- Kada definiramo varijablu, koristimo znak "="
- U ovom slučaju događa se operacija "variable assignment", određenoj varijabli se dodjeljuje određena vrijednost
- Kada se koristi operator "==", događa se druga operacija u kojoj uspoređujemo je li lijeva strana jednaka desnoj
 - Ako je to slučaj, rezultat je True
 - Ako lijeva strana nije jednaka desnoj, rezultat je False

dug = 10

dug = 5

print (dug == 10)

- Što će biti rezultat?

Operatori usporedbe (>, >=, <, <=, ==, !=)

- `x = 10`
- `y = 20`
- `# Jednakost`
- `print(x == y) # Ispisuje False`
- `# Nejednakost`
- `print(x != y) # Ispisuje True`

- `# Manje`
- `print(x < y) # Ispisuje True`
- `# Veće`
- `print(x > y) # Ispisuje False`

Operatori usporedbe (>, >=, <, <=, ==, !=)

Manje ili jednako

```
print(x <= y) # Ispisuje True
```

Veće ili jednako

```
print(x >= y) # Ispisuje False
```

Usporedba stringova

```
a = "banana"
```

```
b = "apple"
```

```
print(a == b) # Ispisuje False
```

```
print(a > b) # Ispisuje True (jer  
'a' dolazi prije 'b' u abecedi)
```

Kako se ustvari koriste operatori usporedbe

- Operatori usporedbe se u stvarnosti koriste u if petljama (o njima poslije)

```
if x > 0:
```

```
    print("Broj je pozitivan.")
```

```
else:
```

```
    print("Broj je negativan ili nula.")
```



Bool tip podataka

- Bool (kratica od Boolean) je temeljni tip podataka u Pythonu koji predstavlja logičku vrijednost, odnosno istinitost nekog izraza
- Može imati samo dvije moguće vrijednosti:
 - True: Označava da je izraz istinit
 - False: Označava da je izraz neistinit
- Jedan od načina korištenja bool tipova podataka je kao rezultat usporedbe dvije vrijednosti
- Bool tip podataka je temeljni koncept u programiranju i omogućuje nam da stvaramo programe koji mogu donositi odluke i reagirati na različite situacije.

Zadatak 3

- Definirajte sljedeće varijable i dodijelite im neke proizvoljne vrijednosti
- Sati_dan
- Sati_noc
- Dnevna_cijena_sata
- Nocna cijena sata
- Izračunajte prvo koliko se zaradi radeći danju, a koliko noću te korištenjem operatora usporedbe je li noćni rad bolje plaćen

Komentiranje koda

- Zašto koristimo komentare u kodu
 - Povećanje čitljivosti koda
 - Dokumentacija koda
 - Olakšavanje održavanja koda
 - Suradnja u timu
- Načini komentiranja
 - Prije koda
 - Odmah pored linije koda



Komentiranje koda - primjeri

```
# podatci o zaposlenicima  
emp_name = "Petar" # employee name  
id = 100 # employee id
```

```
"""
```

```
U ovome modulu smo definirali  
osnovne podatke o našim  
zaposlenicima, za potrebe odjela  
računovodstva
```

```
"""
```

```
print (emp_name, id)
```

Pseudokod

- Pseudokod je način opisivanja kako će računalni program raditi, ali bez korištenja sintakse programskog jezika
- Koriste se jednostavne riječi i rečenice
- Planiranje: Prije nego što se počne pisati pravi program, pseudokod pomaže pri razmislišljanju o svim koracima koje program treba izvršiti.
- Komunikacija: Može se lako objasniti drugim ljudima, čak i onima koji ne znaju programirati, kako program funkcionira.
- Dokumentacija: Pseudokod može poslužiti kao dobra dokumentacija programa, čak i ako ga se kasnije mijenja

Pseudokod primjer – koristimo komentare

```
# korisnik odabire uslugu (auto kredit, stambeni kredit, gotovinski  
kredit)
```

```
...
```

```
...
```

```
# korisnik odabire iznos i rok otplate
```

```
...
```

```
...
```

```
# izračun anuiteta korisniku
```

```
...
```


Sažetak lekcije

- Upoznali smo se s tipovima podataka integers, float, bool
- Integer se koristi za cijele brojeve
- Float se koristi za decimalne brojeve
- Bool se koristi za logičke izraze True i False
- Upoznali smo se s aritmetičkim operatorima +, -, *, /, **, //, %
- Upoznali smo se s operatorima usporedbe
- Naučili smo kako koristiti komentare u kodu - #

