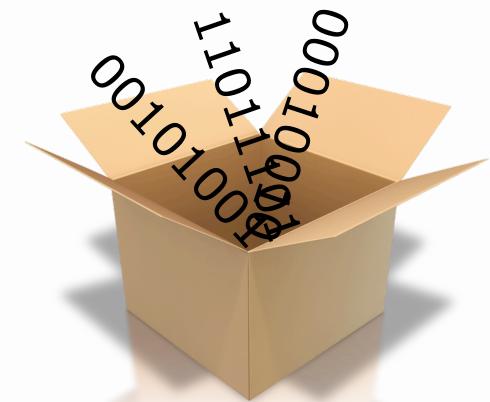




Distance Learning System



Promenljive i tipovi podataka

Python and programming fundamentals

Radna memorija i memorijske adrese

- Memorija je prostor u koji možemo smeštati vrednosti
- Možemo je tretirati kao mnoštvo kutija



- Memorijske adrese su predstavljene brojevima



Radna memorija i memorijske adrese

- Na svaku memorijsku adresu, možemo da stavimo neku vrednost



- Da bismo kasnije mogli da uzmemo vrednosti, moramo znati njihovu adresu
- Odgovorimo na nekoliko pitanja u kontekstu slike iznad
 - Koja je vrednost na adresi 104?
 - Na kojoj se adresi nalazi broj 22?
 - Koliko je $104+105$?

Promenljive / varijable

- Promenljive su simboli kojima se označavaju memorijske adrese
- Zahvaljujući njima, ne moramo pamtitи adrese prilikom pisanja programa
- Promenljivu možemo **deklarisati i inicijalizovati**
- Deklarisanje promenljive možemo posmatrati kao zauzimanje prostora u memoriji:

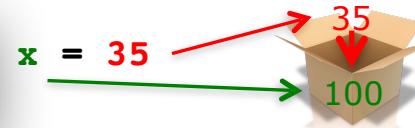


- Inicijalizacija je smeštanje vrednosti u rezervisani memorijski prostor



- U Python-u, deklarisanje i inicijalizacija se obavljaju istovremeno, na sledeći način:

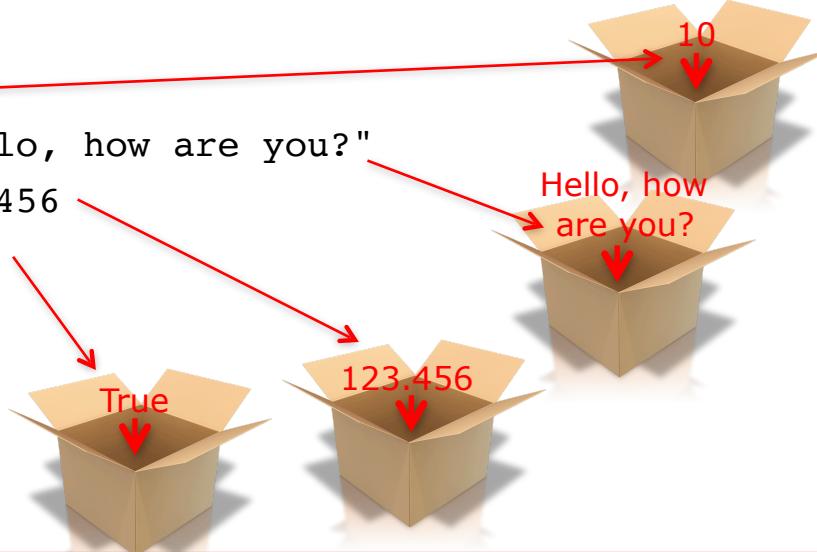
```
myprogram.py > ...
1   x = 35
```



Tipovi podataka

- U radu sa promenljivima, u bilo kom jeziku, najbitnija su dva pitanja:
 - Šta se može smestiti u "kutiju"
 - Koliko toga se može smestiti u "kutiju"
- U Python-u, odgovori na ova pitanja su: **sve** i **sve**, pa možemo napisati:

- `a = 10`
- `b = "Hello, how are you?"`
- `c = 123.456`
- `d = True`



Tipovi podataka

- Iako možemo bilo koju informaciju sačuvati u memoriji, svaka od informacija se tipski klasificuje u jedan od četiri tipa:
 - Ceo broj (**int**)
 - Decimalni broj (**float**)
 - Tekst (**string**)
 - Boolean (**bool**)
- Tip podatka je još jedna informacija koja se čuva u promenljivoj

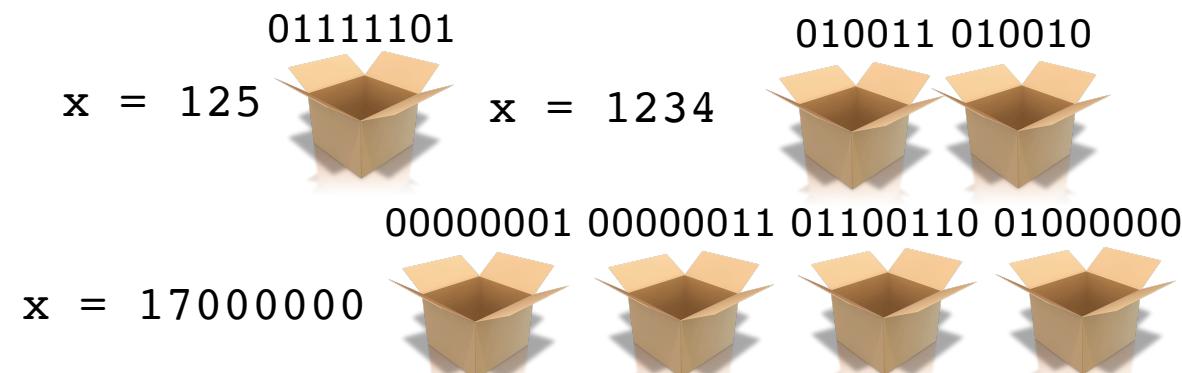
$\text{x} = 123.456$
float



- Ne moramo eksplisitno reći Python-u kog će tipa biti promenljiva – Python će to sam detektovati

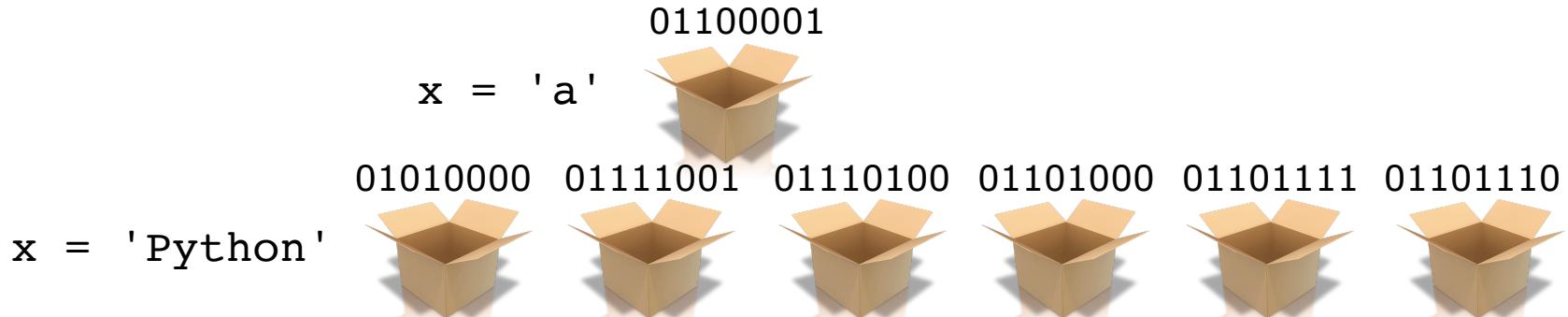
Brojevi

- Svi podaci se u memoriji čuvaju u binarnom obliku (pomoću nula i jedinica)
- Da bismo smestili u memoriju jedan broj, potrebno je da zauzmemos određenu količinu nula i jedinica
- Minimalno zauzeće je jedan bajt, odnosno 8 pozicija za nule i jedinice (što je ukupno 2^8 kombinacija). Sa tim brojem kombinacija, možemo zapamtiti broj od 0 do 256. Za veći broj, treba nam još bajtova (2 bajta su 2^{16} , što je 0 do 65536), sa tri bajta imamo 2^{24} , sa četiri bajta 2^{32} i tako dalje



String

- Tekst se u memoriju takođe smešta u binarnom obliku. Identično kao i brojevi
- Brojevi se kasnije, uz pomoć programa, crtaju na izlazu, kako bi se dobila slova



- Za jedno slovo treba jedan bajt u memoriji. Odnosno, nekada je bilo tako. Ali je ispalo da ima mnogo više slova nego što je moguće smestiti u jedan bajt, pa tako danas i ovaj gospodin 😊 (koji se zvanično delkarise kao karakter) zahteva svoj broj, koji nije baš mali:

11110000 01011111 10011000 10000001

x = "😊"



Boolean

- Boolean nije zahtevan sa stanovišta memorije
- On u sebi nosi uvek samo jednu od moguće dve vrednosti – Da ili Ne
- Da ili ne, u Python-u se označava vrednostima: True ili False

x = True



00000001

y = False



00000000

Još malo o memoriji

Često se sadržaj memorije ne predstavlja binarnim ili decimalnim formatom već na sledeći način



Ovo je samo heksadecimalni format vrednosti jednog bajta



=



=



Imenovanje promenljivih

- Prilikom davanja naziva promenljivima, treba obratiti pažnju na sledeće:
 - Promenljive treba da budu nazvane intuitivno
 - `username`, `pos_x`, `height`, `userBalance`
 - `un`, `px`, `h`, `ub`
 - Promenljive u nazivu ne smeju da imaju prazan karakter (space), niti specijalni karakter i ne smeju početi brojem
 - `firstname`, `user1`, `last_name`, `numberOfBullets`
 - `first name`, `1user`, `user$balance`
 - Promenljive ne smeju biti imenovane isto kao i neka od ključnih reči jezika
 - `userClass`, `willContinue`
 - `class`, `continue`
 - Promenljive su osetljive na velika i mala slova

```
userName = "Peter"
```

 - `print(userName)`
 - `print(username)`

Imenovanje promenljivih

- Kada radimo sa celim brojevima, možemo eksplisitno navesti notaciju kojom želimo da predstavimo vrednost neke promenljive
- Sistem prepoznaće četiri varijante: **binarnu, decimalnu, oktalnu i heksadecimalnu**

a = 10

b = 0b1010

c = 0o12

d = 0xA

- Sve četiri promenljive, nakon izvršenja, imaće vrednost 10

Tipovi pisanja promenljivih

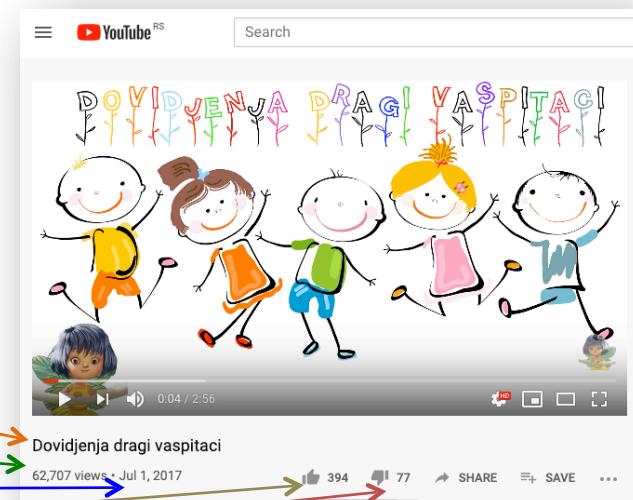
- Osim osnovnih pravila pisanja, postoje i konvencije imenovanja koje nisu određene sintaksom ali se najčešće poštuju
- Najpoznatije konvencije su:
 - Camel case
 - Svaka nova reč se označava velikim slovom
 - userName, firstName, passwordField
 - Hungarian case
 - Prvo se skraćenicom označava tip, a zatim se imenuje
 - txtFirstname, btnSubmit, connDatabase, db_Query



Gde, zašto i kako mogu koristiti varijable

- Varijable su jedan od osnovnih gradivnih blokova
- Koristimo ih da bismo zapamtili vrednosti tokom izvršavanja programa, tamo gde nam neke vrednosti trebaju više od jednog puta
- Koristimo ih tako što ih u nekom delu programa inicijalizujemo, a u nekom drugom delu programa prikazujemo, modifikujemo ili kombinujemo sa drugim vrednostima

```
title = "Dovidjenja dragi vaspitaci"  
views = 62707  
...  
date = "Jul 1, 2017"  
...  
likes = 394  
dislikes = 77
```



Gde, zašto i kako mogu koristiti varijable

```
balance = 15100
```

```
health = 100
```

```
...
```

```
...
```

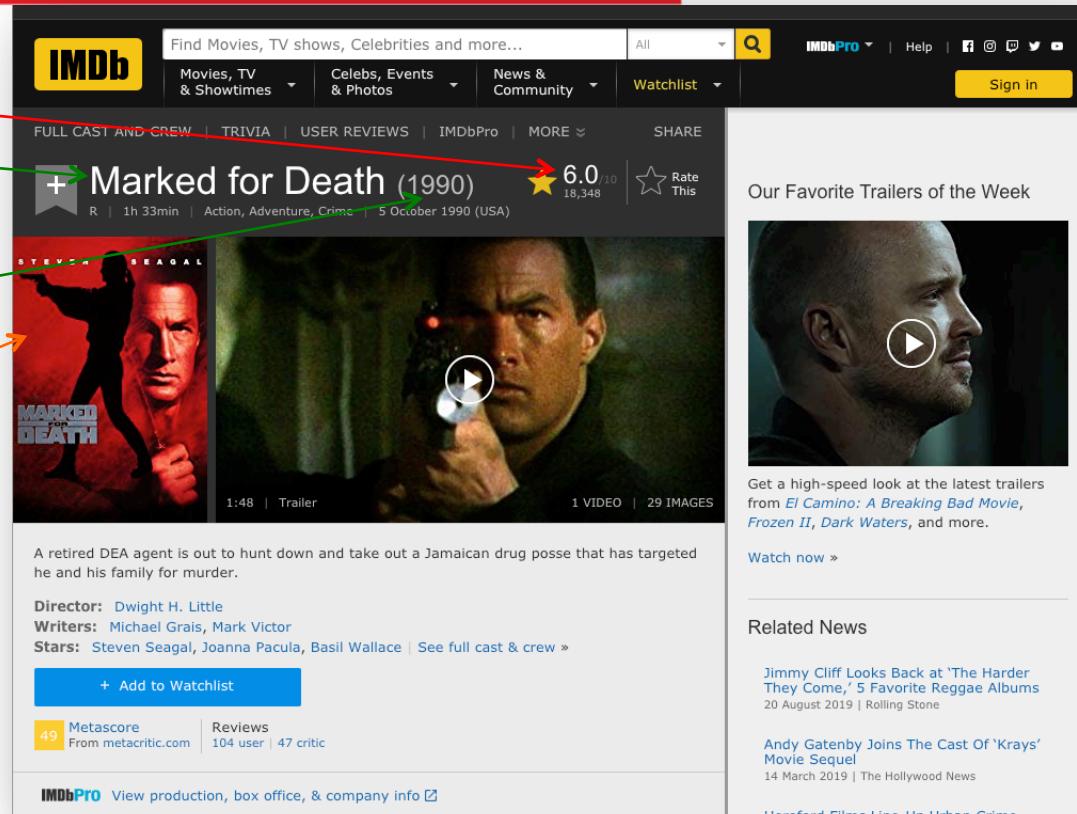
```
armor = 78
```

```
bullets = 6
```



Gde, zašto i kako mogu koristiti varijable

```
grade = 6.0  
...  
title = "Marked for Death"  
...  
...  
year = 1990  
...  
picture = "https://m.media-  
amazon.com/images/M/  
MV5BMzcwNTIzZmEtNzMwMC00OTMyL  
WFfINTQtZjUxZGYzMdk4MzkwXkEyXk  
FqoGdeQXVyMDEwMjgxNg@@._V1_UX  
182_CR0,0,182,268_AL_.jpg"
```



Konverzija tipova

- Ponekad želimo da vrednost nekog tipa pretvorimo u neki drugi tip. Na primer: 1 i "1" imaju istu vrednost, ali prvo je tip int, a drugo string što ove dve vrednosti čini različitima
- Da bismo pretvorili vrednost jednog tipa u drugi tip, koristimo **funkciju** za konverziju
- Na primer:

Ne funkcioniše

```
1 x = "10"  
2 y = 20  
3 print(x+y)
```

Funkcioniše

```
1 x = int("10")  
2 y = 20  
3 print(x+y)
```

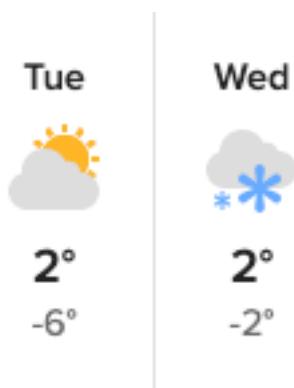
Vežba 1 (ppf-ex03 mob.py)

- Definisati promenljive kojima bi bio predstavljen mobilni telefon (model, proizvođač, slika, cena...)
- Obratiti pažnju na to da vrednosti u promenljivima budu odgovarajućeg tipa
- Prikazati vrednosti na izlazu, komandom print



Vežba 2 (ppf-ex03 forecast.py)

- Definisati promenljive kojima bi bila predstavljena lokalna vremenska prognoza (grad, temperatura, padavine, pritisak...)
- Obratiti pažnju na to da vrednosti u promenljivima budu odgovarajućeg tipa
- Prikazati vrednosti na izlazu, komandom print



Vežba 3 (ppf-ex03 config.py)

- Definisati promenljive kojima bi bila predstavljena konfiguracija računara (naziv procesora, brzina procesora, količina memorije, veličina hard diska, da li se isporučuje sa operativnim sistemom...)
- Obratiti pažnju na to da vrednosti u promenljivima budu odgovarajućeg tipa
- Prikazati vrednosti na izlazu, komandom print



Vežba 4 (ppf-ex03 stats.py)

- Definisati promenljive kojima bi bio predstavljen igrač u RPG igri (ime igrača, health, mana, xp, level...)
- Obratiti pažnju na to da vrednosti u promenljivima budu odgovarajućeg tipa
- Prikazati vrednosti na izlazu, komandom print



Vežba 5 (ppf-ex03 flight.py)

- Definisati promenljive kojima bi bio bila predstavljena pozicija aviona u sistemu za praćenje leta (šifra leta, naziv aviona, latituda i longituda, visina...)
 - Obratiti pažnju na to da vrednosti u promenljivima budu odgovarajućeg tipa
 - Prikazati vrednosti na izlazu, komandom print

