

Vježba – varijable i ispis na ekran

- Kreirajte varijable (imenujte ih i dodijelite im odgovarajuću vrijednost) te ispišite na ekran odgovarajuće vrijednosti, za:
 - Ime, prezime, godinu rođenja, državu rođenja, status radnog odnosa, težinu te spol
 - Stranice a i b, četverokuta te za površinu tog četverokuta.
 - Izračun mjesečne potrošnje el. struje te cijene el. struje koju potroši mikrovalna pećnica snage 1,3 kW ako se koristi 2 sata dnevno?
 - Stranice trokuta, površinu trokuta ($P = \frac{a * v_a}{2}$, v_a je visina na stranicu a) te opseg trokuta.

Vježba

- Kreirajte varijable (imenujte ih i dodijelite im odgovarajuću vrijednost) te ispišite na ekran odgovarajuće vrijednosti, za:
 - Ako automobil troši 5.3 litara na 100 km i ako je cijena goriva 9.56 kn po litri (nije važno kojeg goriva), izračunajte koliko košta 1 km vožnje automobilom. Prikažite mjesečni trošak (30 dana) odlaska na posao automobilom koji je udaljen 20 km u jednom smjeru.
 - Imate 10000 kn i možete zaboraviti na njih na 15 godina. Ako Vam banka nudi 2.5% godišnju kamatu za taj iznos, koliko ćete zaraditi nakon 15 godina. Jednostavni kamatni račun $k = C * p * t$
 - k = iznos kamata odnosno prinos
 - C = iznos glavnice
 - p = godišnja kamatna stopa – NAPOMENA: $5\% = 5 / 100 = \mathbf{0.05}$
 - t = vrijeme u godinama

Unos vrijednosti od korisnika – input()

- Vrijednost varijabli često trebamo dobiti od korisnika našeg programa
- Za unos vrijednosti varijable od korisnika koristi se naredba **input()**

```
ime = input('Upišite ime')
```

```
>>> help(input)
```

Help on built-in function input in module builtins:

```
input(prompt=None, /)
```

Read a string from standard input. The trailing newline is stripped.

The prompt string, if given, is printed to standard output without a trailing newline before reading input.

If the user hits EOF (*nix: Ctrl-D, Windows: Ctrl-Z+Return), raise EOFError.

On *nix systems, readline is used if available.

Vježba – varijable, print() i input()

- Zatražite od korisnika unos dva broja.
 - Nakon unosa brojeva, ispišite:
zbroy, razliku, umnožak, količnik (rezultat djeljnja), potenciranje te modulo
unesenih brojeva
 - Svaka operacija treba biti ispisana u novom redu, a ispis treba imati uključene
brojeve, znak računske operacije te rezultat.
 - PRIMJER ISPIISA:
 - $5 + 8 = 13$
 - $5 - 8 = -3$
 - NAPOMENA Za sada kod unosa neka kod prvog unosa drugi broj NE bude 0
(nula), jer nije dopušteno dijeliti s nulom. To svakako pokušajte napraviti, ali
NE u prvom pokušaju.
- **PROBLEM!!! Dogodila se greška!**

Greške u programskom kôdu

- Ljudi nisu savršeni.
- Kôd koji ljudi napišu nije savršen, ima grešaka.
- Kôd koji napišu roboti, koje su napravili ljudi, također ima grešaka



BUG

Znate li priču o bubi u računalu zbog koje je računalo prestalo raditi i zbog koje se kolokvijalno greške u kôdu nazivaju BUG.

9/9


0800 Antan started
1000 " stopped - antan ✓

1300 (033) MP-MC { 1.2700 9.037847025
1.982647000 9.037846995 connect
2.130476415 (033) 4.615925059(-2)
(033) PRO 2 2.130476415
connect 2.130676415

Relays 6-2 in 033 failed special speed test
in relay .. 11.00 test.

Relays changed

1100 Started Cosine Tape (Sine check)
1525 Started Multi-Adder Test.

1545  Relay #70 Panel F
(moth) in relay.

First actual case of bug being found.

1630 Antan started.
1700 closed down.

Relay 2345
Relay 3376

[First Computer Bug, 1945 - Software bug - Wikipedia](#)

Greške u kôdu – tipovi grešaka

- **Syntax Error** – greške u sintaksi. Krivo napisane naredbe, nazivi funkcija ... Najčešće program NE možete pokrenuti s ovom greškom, ali IDE točno ukazuje gdje se nalazi ovakva greška.
- **Runtime Error** ili **Exception Error** – greške koje nastaju prilikom izvođenja programa. Ponekad ih ne uočite odmah (primjer pokretanje zadnjeg zadatka). Malo ih je teže otkriti.
- **Bugs** – greške koje je najteže otkriti. To su greške koje predstavljaju krivo funkcioniranje programa. Primjer: uz cijenu se ne obračunava PDV, nego su sve cijene iste. Dakle, program funkcionira, kôd je ispravno napisan, ali program ne radi ono što bi trebao.



Konverzija tipova podataka

- Ponekad je nužno napraviti konverziju iz jednog tipa podatka u drugi tip
- Najčešće se konverzija radi iz teksta (string) u broj (integer ili float) ili obrnuto (Primjer kada trebamo vrijednost varijable zapisati u datoteku).
- Postoje i naredbe za konverziju brojeva između brojevnihi sustava – binarni, oktalni, dekadski, heksadekadski

Konverzije tipova podatka

- Integer – **int('string_koji_konvertiramo')** – cijeli broj
- Float – **float('string_koji_konvertiramo')** – decimalni broj,
- String – **str(broj_ili_objekt_koji_konvertiramo)** – pretvara u tekst,
- Boolean – **bool('True')** – string u boolean
OPREZ!!! Ova funkcija će pretvoriti bilo koji string koji ima znakove u True, osim jedne iznimke. To je 'False'.
- Character – **chr(broj_koji_konvertiramo)** – pretvara u Unicode znak
- Unicode Character – **ord('string_koji_konvertiramo')** – pretvara u cijeli broj koji predstavlja Unicode znak

Konverzija između brojevnih sustava

- Dodatne funkcije za konverziju između brojevnih sustava (NE tipova podataka)
- Iz dekadskog u binarni – **bin(cijeli_broj_koji_konvertiramo)**
- Iz dekadskog u oktalni – **oct(cijeli_broj_koji_konvertiramo)**
- Iz dekadskog u heksadekadski – **hex(cijeli_broj_koji_konvertiramo)**
- Iz ... natrag u dekadski – **int('broj_koji_konvertiramo', baza)**
 - Broj koji konvertiramo treba biti tipa string pa zato imamo navodnike
 - Baza predstavlja bazu sustava iz kojeg konvertiramo. Binarni je 2, oktalni je 8, heksadekadski je 16

Vježba – varijable, print(), input(), konverzija

- Prepravite prethodne vježbe i zadatke tako da vrijednosti varijabli tražite od korisnika te napravite potrebnu konverziju.
 - *Otvorite .py datoteke u kojima imate riješene zadatke te ih prepravite tako da koristite i naredbu input(). Zatim ih možete pohraniti pod novim imenom ili ostaviti isti naziv.*
- IP adresa je adresa svakog računala na mreži koja se sastoji od četiri broja između 0 i 256. Primjer IP adrese: 192.168.0.184
 - IP adresu iz primjera ispišite u binarnom, oktalno i heksadekadskom obliku.
 - SAVJET: Za sada koristite zasebnu varijablu za svaki od četiri broja, odnosno dijela (okteta) IP adrese, ali ispišite ih u istom obliku kako je navedeno u primjeru (192.168.0.184).
 - Ispis treba napraviti za sve oblike brojevni sustava.
- Na stranici <https://www.color-hex.com/color-palette/33532> imate boje Google logotipa. Pomoću odgovarajućih naredbi za konverziju pokušajte pretvoriti RGB zapise u HEX boja i obratno.
 - Primjer:
 - Naziv boje HEX zapis RGB (Red Green Blue)
 - CRVENA #EA4335 (234, 67, 53)
 - Za HEX zapis EA-43-35 trebate dobiti RGB zapis 234-67-53
 - NAPOMENA Zanimajte početni # znak u HEX zapisu sa stranice.
 - NAPOMENA HEX zapis čine tri grupe po dva znaka EA-43-35, svaka dva znaka čine jednu boju RGB

Vježba – prepravite ove zadatke...

- ... tako da za unos vrijednosti pitate korisnika:
 - Ako automobil troši 5.3 litara na 100 km i ako je cijena goriva 9.56 kn po litri (nije važno kojeg goriva), izračunajte koliko košta 1 km vožnje automobilom. Prikažite mjesečni trošak (30 dana) odlaska na posao automobilom koji je udaljen 20 km u jednom smjeru.
 - Imate 10000 kn i možete zaboraviti na njih na 15 godina. Ako Vam banka nudi 2.5% godišnju kamatu za taj iznos, koliko ćete zaraditi nakon 15 godina. Jednostavni kamatni račun $k = C * p * t$
 - k = iznos kamata odnosno prinos
 - C = iznos glavnice
 - p = godišnja kamatna stopa – NAPOMENA: $5\% = 5 / 100 = \mathbf{0.05}$
 - t = vrijeme u godinama