Python, teorija

Što je to varijabla ?

Uzmimo za primjer čašu za vodu. U toj čaši se može nalaziti voda, a kada popijete vodu u tu istu čašu možete staviti sok.

Znači u našoj zamišljenoj čaši u nekom trenutku možemo imati razne vrste napitaka, ali nikada više različitih napitaka u istom trenutku. (naravno ako poštujemo pravila).

U informatici i u postupku programiranja naša zamišljena čaša naziva se **varijabla**.

Znači varijabla je neka promjenjiva vrijednost koja se može po potrebi mijenjati i poprimati različite vrijednosti.

Pravila pisanja varijabli:

a) *Varijable obavezno pišemo malim slovima engleske abecede*
b) *Varijabla može biti i jedna riječ*
c) *Želite li koristiti dvije ili više riječi obavezno ih odvojite donjom crtom npr. petar_peric*
d) *Varijabla ne smije početi sa brojem*

Analizirajmo naš program, ali samo ona slova koja su označena crvenom bojom. **To su naše varijable.**

Program
a=3 b=5 c=a+b print("Rezultat zbrajanja je",c)

Varijabli a pridružili smo broj 3, a varijabli b broj 5.

Varijabla c nam služi da se u nju pohrani rezultat zbrajanja vrijednosti u varijabli a i vrijednosti u varijabli b.

Kada želimo ispisati rezultat zbrajanja naših varijabli, moramo prikazati vrijednost varijable c jer se u njoj nalazi rezultat zbrajanja.

Brojevi u varijablama ostaju pohranjeni sve dotle dok je program pokrenut ili ne u nesemo neku novu vrijednost.

Brojevi koji su pohranjeni u varijablama mogu nam poslužiti da sa tim brojevima napravimo razne matematičke operacije .

Pogledajmo primjer:

Program	**Pojašnjenje**
a=3 b=5 c=a+b d=a-b e=a*b print("Rezultat zbrajanja je",c) print("Rezultat oduzimanja je" ,d) print("Rezultat množenja je ", e)	U našem primjeru dodali smo još dvije varijable (d,e), naravno u varijabli d će biti pohranjen rezultat oduzimanja, a u varijabli e bit će pohranjen rezultat množenja. Da bi smo ispisali vrijednost neke varijable moramo koristiti naredbu print. To je ujedno i naša prva naredba za učenje.

Aritmetičke operacije u Python_u

Osnovne matematičke operacije	Koristit ćemo brojeve 8 i 3 – za primjer:
+ zbrajanje	8+3 = 11
-oduzimanje	8-3 = 5
* množenje	8*3 = 24
/ djeljenje	8/3 =2.6666
// cjelobrojno dijeljenje	8//3 =2 - cijeli broj
% ostatak cjelobrojnog dijeljenja	7%3 =1 - ostatak dijeljenja (2 i ostatak 1)

Vrlo važno: proučite sljedeće primjer:

Proučite sljedeće primjer			
16%4= 0 6%6= 0 6%1 =0 6%2 =0 6%3=0 6%4=2 6%5=1	5%6=5 4%6=4 5%6=5	16//4=4 16//5=3 16//4=4 16//3=5 16//2=8 16//1=16 16//16=1	4//16=0 5//16=0 12//16 =0

Pretvaranje matematičkih operacija u Python kod

a2+3b2 =ab	a*2+3*b*2=a*b	Važno : **Matematički zapis** **2b+3c** **U Pythonu obavezno mora biti u ovakvom zapisu:** **2*b+3*c**
$\frac{a+b}{2x}$	(a+b)/(2*x)	
8:4+2:2-3b	8/4+2/2-3*b	
6-6a+2b	6-6*a+2*b	
U matematici znamo redoslijed matematičkih operacija : **a) Zagrade b) Množenje ili dijeljenje c) Zbrajanje ili oduzimanje**		
Važno pravilo : *Kod operacija istih vrijednosti obavezno obratite pažnju da se operacije vrše s lijeva na desno:* **npr: 2*(6/2)/2*3 = Što je rezultat ? 9 ili 1 Odgovorite??**		

Naredba PRINT

Pravilo pisanja naredbi , naredbe kao i varijable obavezno moraju biti pisane malim slovima.

Kao što smo vidjeli naša naredba print služi za ispis pohranjene vrijednosti u **varijablama**.

Naredba **print** ima i svoja pravila

1. Tekst koji želimo ispisati na ekranu obavezno mora biti pod navodnicima, ono što je pod navodnicima kako je napisano tako će se prikazati na ekranu, možemo koristiti bez ikakvih problema mala i velika slova kako god mi želimo.
2. Varijable ne smiju biti pod navodnicima ,
3. Naredbom **print** možemo urediti prikaz rezultata operacija nad varijablama.

Primjer_1.

Program	**Ispis na ekranu , vrijednosti varijabli su 3 i 2**
a=3 b=2 c=a+b print("Rezultat zbrajanja je",c) print("Rezultat zbrajanja broja",a,"i broja",b,"je",c) print(c)	Rezultat zbrajanja je 5 Rezultat zbrajanja broja 3 i broja 2 je 5 5

Koji je ispis ljepši ???? – prokomentirajmo !!

Naredba print može u sebi sadržavati razne matematičke operacije što će vam nekada dobro doći.

Program	**Objašnjenje**
a=3 b=2 print("Rezultat zbrajanja je",a+b)	U ovom primjeru nemamo varijablu za smještanje rezultata zbrajanja kao u gornjem primjeru. Operacija zbrajanja je u naredbi print i odmah će se ispisati , vrijednost zbrajanja se ne može koristiti u daljnjem programu.

Naredba **print** koristi još dva dodatna znaka /n za prelazak u novi red te /t tabulator za odvajanje riječi.

Primjer:

print("Pozdrav \n" , "Učimo Python\n")	Pozdrav Učimo Python
Print("Pozdrav\t","Učimo Python\t")	Pozdrav Učimo Python

Naredba INPUT

Naredba INPUT nam služi za upis vrijednosti u varijablu, ***prisjetite se da smo rekli da naredba print služi za ispis vrijednosti varijable.***

Pravila za naredbu input:

Pošto naredbu input koristimo za unos vrijednosti u varijablu tj. preko tipkovnice možemo unijeti slova ili brojeve.

*U naredbi **input** moramo naznačiti što unosimo.*

Naredba input obavezno počinje s nekom varijablom kako bi u nju mogli pohraniti neku vrijednost koju unosimo preko tipkovnice.

Kod unosa brojeva moramo naznačiti da li se radi o cijelom broju ili decimalnom broju.

Za cijeli broj koristimo naredbu : int
Za decimalne brojeve koristimo naredbu : float
Za unos teksta naredba input *nema nikakvih dodataka*

Program	
a=int(input("Unesite prvi broj ")) b=float(input(" Unesite drugi broj")) x=input("unesi svoje ime") c=a+b print("Rezultat zbrajanja je",c)	1. Cijeli broj npr. 2,4 6,10,101. 2. Decimalni broj 2.3 ,3.555 ,5.54 3. Običan tekst **Kod decimalnih brojeva ne koristimo zarez, koristimo točku.** ***Važno – koliko ste zagrada otvorili toliko zagrada morate i zatvoriti.***

UVJETNE NAREDBA IF-ELSE—ELIF

Što moramo znati??: naredba za ispis je print(), "=" je znak pridruživanja vrijednosti varijabli. Postoji više načina ispisa (poželjno da znate barem jedan), a aritmetičke operatore ste obavezni znati.

Dosad smo u programima koristili aritmetičke operatore. Međutim, što učiniti ako trebamo u programu uspoređivati više vrijednosti ?

Tu nam pomažu operatori uspoređivanja.

Ima ih nekoliko, također ih već znate iz matematike, ali u matematici se oni pišu malo drugačije u odnosu na Python. U nastavku se nalazi prikaz operatora.

Značenje	**Oznaka u matematici**	**Oznaka u Pythonu**
Manje od	**<**	**<**
Veće od	**>**	**>**
Manje ili jednako	**≤**	**<=**
Veće ili jednako	**≥**	**>=**
Jednako	**=**	**==**
Različito	**≠**	**!=**

NAPOMENA: Pazite na razliku između "=" i "==". **Znak** "=" je znak pridruživanja koji služi da se neka

vrijednost pridruži varijabli, dok znak "==" predstavlja operator uspoređivanja koji je istinit ako su uspoređene vrijednosti jednake.

Možemo dalje ?. Naime, u programu mogu postojati uvjeti u kojima se program grana na više mogućih rješenja ovisno o uvjetu.

Primjer_1:

Evo primjera iz stvarnog života; dođemo u dućan sportske opreme i kupujemo sve što nam treba za dječje igralište . Na kraju kupnje dođemo na blagajnu i u slučaju iznosa većeg od 5000 kn imamo pravo na besplatnu dostavu što se ispisuje na računu. Upravo ćemo ovaj zadatak riješiti, ali ćemo ga malo pojednostaviti. Kako ne bismo unosili svaku stavku posebno, unijet ćemo samo konačnu vrijednost kupovine, a program nam treba ispisati: vrijednost računa, pravo na dostavu (samo u slučaju vrijednosti računa većeg od 5000 kn) te prigodnu rečenicu ("Dođite nam opet!").

Sintaksa za if

If uvjet:
blok naredbi

#Unos vrijednosti računa
racun=float(input("Unesite vrijednost računa"))
#Uvjet if i ispis prigodne rečenice
if racun>5000:
  print("Imate pravo na besplatnu dostavu")
print("Dođite nam opet")
	# - koristimo za komentare ,oni služe za lakše snalaženje u programu i ne utječu na izvršavanje programa.
	Ono što je u uvjetu if ,mora biti uvučeno, najbolje koristite tipku tab na tipkovnici.
	Ovaj dio naredbe nije vezan za uvjet if

Kao što vidite na primjeru, if uvjet sastoji se od ključne riječi **if**, uvjeta i nakon toga slijedi znak ":".
Važno je nakon toga naglasiti da sve što spada pod naredbu IF se nalazi uvučeno i to će se izvršiti **samo ako je uvjet zadovoljen**.

Kao uvlaka se preporučuje tipka tab i to uređen kao četiri razmaka.

Naravno, to je najjednostavnije korištenje if uvjeta. Sada zamislimo sljedeći slučaj. Došli smo u školu i trebamo napraviti program koji će nam ispisivati da li smo prošli ili pali ispit. To ćemo učiniti tako da napravimo program koji će u slučaju ocjene 1 ispisivati "Niste položili ispit.", a za ostale slučajeve ispisivati "Položi ste ispit ! Čestitamo"

Sintaksa za if – else:

```
if uvjet:
    blok naredbi
else:
    blok naredbi
```

# unos ocjene **ocjena=int(input("Unesite ocjenu "))** **#Uvjet za if – else** **if ocjena==1:** **print("Niste položili ispit.")** **else:** **print("Položili ste ispit ! Čestitamo")**	**Kao što vidite u uvjet if koristimo znak jednakosti ==.** **Ako je uvjet točan izvršit će se onaj dio programa koji je u uvjetu if.** **Ako uvjet nije točan izvršit će se onaj dio programa koji se nalazi ispod else. *Važno* je zamijetiti da se naredbe ispod if i else moraju uvući.**

Programski primjeri s rješenjima:

Napiši program koji unosi dva broja te ispisuje poruku jesu li brojevi jednaki ili različiti.	*#program za unošenje i usporedbu dva unesena broja* **a=int(input("Upiši prvi broj:"))** **b=int(input("Upiši drugi broj:"))** **if a==b:** **print("Brojevi su jednaki")** **else:** **print("Brojevi su različiti")**

Primjer_2:

Napiši program za unos dva broja (input) te ispisuje poruku koji od ta dva broja je veći.	**#usporedba unesenih brojeva** **a=int(input("Unesi prvi broj:")** **b=int(input("Unesi drugi broj:")** **if a>b:** **print("broj",a,"je veći")** **else:** **print("broj",b,"je veći")**

Primjer_3:

Napišite program koji unosi duljine stranica a i b te ispisuje poruku da li se o kvadratu ili pravokutniku. **Važno:** **Što je karatteristično za kvadrat ??**	**a=int(input("duljina stranice a:"))** **b=int(input("duljina stranice b:"))** **if a==b:** **print("KVADRAT")** **else:** **print("PRAVOKUTNIK")**

Kao što možete vidjeti, za uspješno rješavanje zadataka sa uvjetom IF – ELSE bitno je poznavati matematičke operatore uspoređivanja.(>,<,==,!=..) i aritmetičke operatore (+,-,*,/, //, %.)..

Sintaksa za if – elif – else: *višestruki uvjeti:*

If uvjet:

blok naredbi

elif uvjet:

blok naredbi

else:

blok naredbi

Zamislimo situaciju da ovisno o unesenoj ocjeni od 1 do 5, program ispisuje poruku ocjenu koju ste dobili i da li ste ili niste prošli zamišljeni ispit.

```
# unos ocjene
ocjena=int(input("Unesite ocjenu:  "))
# uvjet if – elif – else
If ocjena==1:
    print("Ocjena ispita je nedovoljan, niste prošli")
elif ocjena==2:
    print(" Ocjena ispita je dovoljan, prošli ste")
elif ocjena==3:
    print("Ocjena ispita je dobar,prošli ste")
elif ocjena==4:
    print("Ocjena ispita je vrlo dobar,prošli ste")
elif ocjena==5:
    print("Ocjena ispita je odličan, prošli ste")
else:
    print(" Unijeli ste krivu ocjenu")
```

elif – kao što možete vidjeti na ovom primjeru koristimo višestruke uvjete za prikaz svake ocjene od 1 do 5.

Ako ocjena nije u rasponu od 1 do 5 aktivirat će se događaj pod naredbom else.

UVOD U Petlje

Prošli smo put radili uvjete, odnosno grananje programa u ovisnosti o nekom uvjetu ili stanju u programu. Danas ćemo obraditi (petlje, ponavljanje dijela programa), koje su vrlo jednostavne.

Što je uopće petlja i čemu služi?

Petlja predstavlja dio programa koji možemo ponoviti određen broj puta. Na taj si način olakšavamo posao i nemamo potrebe pisati neki dio programa više puta.

Najjednostavniji primjer petlje je onaj školski; napišite 5 rečenica „Volim programiranje!“. Iako je to relativno mali broj ponavljanja, nekima neće biti problem i napisat će pet redova i u svakom ispisati rečenicu s pomoću funkcije print().

Međutim što ako umjesto broja 5 piše broj 50?

Naravno, možete se koristiti copy/paste metodom i brojiti redove, ali čemu se mučiti i filozofirati? Upravo zato upotrebljavamo petlje.

Postoji više vrsta petlji, a mi ćemo naučiti dvije (for i while) petlje, odnosno petlju s brojačem (*for*) i petlju s uvjetom (*while*).

Prije nego što krenemo rješavati zadatke, hajde da na ovom prethodnom primjeru prikažemo upotrebu petlji. Prvo da vidimo kako bi to izgledalo s pomoću *for* petlje:

```
for brojac in range(1,6):
        print(brojac)
```

Na početku vidimo ključnu riječ *for*, koja nam objašnjava da se radi o petlji s brojačem.

Nakon toga vidimo **VARIJABLU** brojac koja će poprimati vrijednosti iz funkcije range(). Unutar funkcije range() vidimo vrijednosti 1 i 6 odvojene zarezom. Ovdje treba napomenuti da je prva vrijednost početna, a druga završna, s tim da se ponavljanje izvršava do završne, **a nju ne uključuje**.

U ovom bi primjeru varijabla brojac u prvom koraku imala vrijednost 1 i zatim bi se s pomoću funkcije print() ispisao broj 1. Nakon toga se opet vraćamo na početak petlje i varijabla ima vrijednost 2 te se opet to ispisuje. I tako sve do vrijednosti 6, gdje brojač dobiva upravo tu vrijednost. Budući da prema funkciji range() taj broj ne ulazi u interval, ne ulazi se u petlju, nego se izlazi van. I na taj način smo ispisali brojeve od 1 do 5.

Kod petlje s uvjetom, odnosno petlje while, obavezni smo sami napraviti brojač jer se u zaglavlju petlje zapravo ispisuje uvjet, to jest, u ovom slučaju, je li naš brojač manji od 6, a to izgleda ovako:

```
brojac = 1
while brojac < 6:
    print(brojac)
    brojac = brojac + 1
```

Za početak, prije same petlje, moramo napraviti varijablu brojac u kojoj se nalazi naša početna vrijednost, u ovom slučaju to je broj 1. Nakon toga slijedi ulazak u petlju. Vidimo ključnu riječ while, koja označuje početak while petlje, a nakon toga slijedi uvjet gdje se u slučaju ako je uvjet zadovoljen ulazi u petlju, a u suprotnome izlazi iz nje.

Pri prvom koraku petlje vrijednost brojača je 1 i to je manje od 6, pa se ulazi u petlju i ispisuje vrijednost brojača. Nakon toga se brojaču povećava vrijednost za 1.

Ovo je vrlo važno jer bi u protivnom vrijednost brojača ostala 1, tj. bila bi konstanta i u tom slučaju bi **uvjet** uvijek bio zadovoljen i petlja bi se stalno izvodila što nazivamo beskonačnom petljom. Ipak, ovdje smo povećali vrijednost varijable brojača koja se izvršava sve dok je brojač manji od 6. Kada vrijednost brojača dođe do broja 6, dolazimo do zaglavlja petlje, ali uvjet nije ispunjen pa se zbog toga izlazi iz petlje.

PETLJA FOR

Kao što smo do sada naučili to je konačna petlja koja ima svoj početak i kraj.

Petlja for ima više oblika i prikazat ćemo sve mogućnosti.

U petlji for brojevi i rezultati obrade uvijek se ispisuju u stupcu tj. jedan ispod drugoga

Oblik_1: Imamo definiranu početnu i završnu vrijednost

for a in range(1,6): print(a)	a – predstavlja brojač koji nam poprima vrijednost za jedan više svaki puta kada prolazi kroz petlju do konačno broja 6. koji je naveden u funkciji range(1,6). Naredbe koje nalaze u petlji obavezno moraju biti uvučene, to smo učili kod uvjeta.

Oblik_2: Definiramo početnu vrijednost, završnu vrijednost i korak uvećanja

for a in range(1,6,2): print(a)	(1,6,2) – Prvim prolaskom kroz petlju ispisat će se broj 1, drugim broj 3, sljedećim prolaskom broj 5. Ispisat ćemo neparne brojeve od 1 do 6. Što ako želimo ispisati parne brojeve ? Stavit ćemo samo range(0,6,2)

Oblik_3: Ispis brojeva od većeg prema manjem

for a in range(6,1,-1): print(a)	(6,1,-1) – plavi -1 – predstavlja da idemo u natrag od većeg prema manjem(5,4,3,2,1) Razmislite !! Što bi ste promijenili da ispišete neparne brojeve od 6 do 1, a što da ispišete parne brojeve od 6 do 1)??

Oblik_4: Ispis brojeva u retku

for a in range(1,6): print(a,end=" ")	end="" – omogućuje ispis rezultata obrade u retku. Važno: ako stavite razmak između navodnika, dobit ćete i lijepi razmak između brojeva u retku . Stavite li između navodnika (- ili ,) brojevi će biti međusobno odijeljeni tim znakom.

Oblik_5: Petlja for i zamjena naredbe range sa in i not in

a="sunce" for i in a: print(i)	Umjesto range , koristimo ključnu riječ in. In – provjerava da li se ono što tražimo nalazi u stringu.(znakovnoj varijabli).