Programiranje za web

Nastavna jedinica: Python: For petlja i stringovi

prof. Dominik Marijačić

Gradivo izvodite u online Python editoru, nalazi se [ovdje](https://www.onlinegdb.com/online_python_compiler).

**For petlja**

Pošto vas nije bilo dosta prošli put, ponovit ćemo koncept for petlje u Pythonu. Na sljedećoj slici vidite najjednostavniji primjer petlje:

For petlja u Pythonu se sastoji od par glavnih djelova.

* prvi dio je varijabla i koja ima ulogu brojača unutar petlje.
* drugi dio je range, ili raspon (hrv.). Raspon petlje se sastoji od 3 dijela.
* prvi broj unutar raspona je 1, a taj broj označava početak brojača, tj. početnu vrijednost brojača.
* to znači da brojač počinje brojiti od broja 1
* drugi dio petlje je kraj brojača, tj. vrijednost iznad koje brojač više ne ide. U ovom slučaju, brojač broji do 40, točnije do 39, jer broji sve dokle je i < 40; dakle brojač broji sve dok je manji od toga broja (40)
* treći dio raspona je stupanj promjene. Taj broj nam govori za koliko se brojač promijeni nakon svakog izvršavanja petlje. U ovom slučaju brojač se poveća za 1, zato što je broj koji je na trećem mjestu raspona broj 1.
* Sve što se nalazi unutar petlje (ispod for i in range(1,40,1):) je ono što petlja izvršava svakim svojim izvršavanjem.
* U ovom slučaju to je naredba print(i); drugim riječima, petlja svakim svojim izvršavanjem ispisuje vrijednost brojača

Kakav će biti ispis ove petlje?

Kao što je rečeno, petlja će ispisivati vrijednost brojača, svakim svojim izvršavanjem. To znači da će ispis biti brojevi od 1 do 39, nešto ovako:

1

2

3

.

.

.

39

**Sljedeći primjeri:**

*Neka petlja ispiše sve brojeve u rasponu od -15 do 45 koji su neparni.*

U ovom slučaju petlja provjerava svaki broj od -15 do 45. To znači da je stupanj promjene petlje 1, dakle treći broj u rasponu petlje je 1. Drugim riječima, nakon svakog prolaska petlje, brojač se povećava za 1.

Dakle, petlju definiramo ovako:

Dakle, početna vrijednost brojača petlje je -15. To je broj odakle brojač počinje. Kraj brojača je 46. Ako se pitate zašto 46, a ne 45, to je zato što brojač broji sve dok je manji od 46. Dakle, brojač ide do 45 i tu staje. Treći dio petlje je logičan, to je broj 1; to znači da se brojač svakim prolaskom povećava za 1.

Kako provjeriti je li broj neparan?

Kada nešto provjeravamo, koristimo uvjete (if). U ovom slučaju provjeravamo je li ostatak djeljenja broja sa 2 jednak nuli ili nije jednak nuli.

Neparni brojevi imaju ostatak različit od nule kada se podijele sa 2. To izgleda ovako:

Ispis ovog programa izgleda ovako:

-15

-13

.

.

.

45

**For petlje sa stringovima**

Stringovi su varijable tekstualnog tipa. Stringovi imaju svoju duljinu, isto kao što svaka riječ ima svoju duljinu. Duljina stringa može poslužiti kao raspon u for petlji.

Pogledajmo sljedeći primjer:



Kao što vidimo, imamo neku varijablu a. Ta varijabla je string, jer joj je vrijednost riječ nogomet. Varijabla a ima svoju duljinu. Koja je duljina varijable a?

Duljina varijable a je 7, zato što riječ nogomet ima 7 slova.

Ispis duljine stringa a se postiže sa: len(a). len(a) je, dakle, 7.

Kako to možemo upotrijebiti u for petlji?

Pogledajmo sljedeći primjer:

U ovom slučaju, početak brojača je 0, a kraj brojača je len(a), ili 7. (len(a) = 7)

Svakim prolaskom petlja ispisuje vrijednost a[i]. Dakle, ako je i = 0, petlja ispisuje vrijednost a[0], što je slovo „n“. Sljedećim prolaskom i se povećava za 1 pa je i = 1.

Tada petlja ispisuje vrijednost a[1], što je slovo „o“.

Petlja, dakle, ispisuje slovo po slovo varijable a.

n

o

g

o

m

e

t

Prisjetimo se primjera koji smo često rješavali, a to je: Program koji broji koliko slova a ima vaše ime.

Pogledajmo sljedeći primjer:

Analizirajmo program, red po red.

* u prvoj liniji koda definiramo string ime i dajemo mu vrijednost imena. U ovom slučaju radi se o imenu „marijana“.
* u drugoj liniji koda definiramo posebni brojač koji će se povećati za 1 svaki put kada detektira slovo a u stringu.
* nakon toga definiramo for petlju. Ova for petlja ima raspon od 0 do duljine stringa ime. U ovom slučaju string ime ima duljinu 8 (len(ime) = 8). Dakle, petlja ide od 0 do 8 (Dakle ide od 0 do 7 jer se izvršava sve dok je i <8)
* unutar for petlje provjeravamo sa if uvjetom.
* ako je ime[i] (dakle slovo kojeg petlja promatra) jednako slovu a:
* povećaj brojač za 1 (br = br+1)
* kada se sve završi, program ispisuje: Ime marijana ima 3 slova a.

Dakle, petlja gleda slovo po slovo stringa ime. Počinje sa brojačem i = 0 i gleda je li ime[0] jednako slovu a. ime[0] je slovo m, tako da se petlja opet izvršava, sada sa i = 1. ime[1] je jednako slovu a, prema tome brojač br se povećava za 1.

Tako petlja prođe kroz svako slovo imena i na kraju ispisujemo vrijednost brojača br, a to je 3. Toliko je slova a u imenu „marijana“.

Sljedeći primjer:

Neka program pošalje upit za unos imena, i nakon toga pozovite funkciju koja provjerava je li duljina vašeg imena neparan broj. Ako jest, neka program ispiše „zgoditak“, ako nije, neka program ispiše „gubitak“.

Program bi trebao izgledati ovako:

Ako u programu radimo s funkcijama, prvo treba definirati i isprogramirati funkciju.

* Prvo se definira funkcija imena f kojoj pridružujemo parametar ime
* Funkcija ima jedan parametar zato što je funkciji potrebna samo jedna informacija da bi izvršila ono što se traži od nje
* Od funkcije se traži da provjeri je li duljina vašeg imena paran ili neparan broj; dakle, funkciji treba poslati samo jedan string, string koji sadržava vaše ime
* Pošto funkcija provjerava, u funkciji trebamo postaviti uvjet (if).
* Funkcija provjerava je li ostatak dijeljenja duljine vašeg imena različit od nule. Ako jest, to znači da je duljina vašeg imena neparna, prema tome ispisuje se „zgoditak“
* Ako taj uvjet nije ispunjen, izvršava se ono pod else, dakle piše se „gubitak“
* Nakon funkcije, u glavnom programu definiramo varijablu a čiju vrijednost upisujemo u konzolu (odatle ovo input)
* Nakon toga, funkcija se poziva: f(a) ---> f je ime funkcije, a *a* je varijabla u koju smo upisali svoje ime

**ZADATCI ZA DOMAĆU ZADAĆU**

1. Napravi petlju koja ispisuje sve brojeve djeljive s 3 od 77 do 45.
2. Definiraj varijablu p, tipa string, koja ima vrijednost „insuficijencija“. For petljom provjeri koliko ima slova i u tom stringu. Ispis bi trebao biti ovakav:

*Riječ insuficijencija ima 4 slova i.*

1. Napravi funkciju koja provjerava je li vaše prezime veće duljine od duljine vašeg imena. Program treba poslati upit za unos imena i zatim upit za unos prezimena. Nakon toga pozivate funkciju koja provjerava je li prezime dulje od imena.

Ako je to slučaj, program treba ispisati:

*Moje prezime je dulje od mog imena za \_\_\_ slova.*

Ako je vaše ime veće duljine od vašeg prezimena, program treba ispisati:

*Moje ime je dulje od mog prezimena za \_\_\_ slova.*

Ako su vaše ime i prezime jednakih duljine, program treba ispisati:

*Moje ime i prezime su jednakih duljina.*

 Ako ima ikakvih nedoumica i upita, pošaljite poruku u grupu!