

Python programski jezik

Varijabla je memorijska lokacija simboličnog imena u koju se sprema vrijednost nekog podatka. Varijabla se sastoji od imena, adrese i vrijednosti.

Glavni postulati varijable su sljedeći:

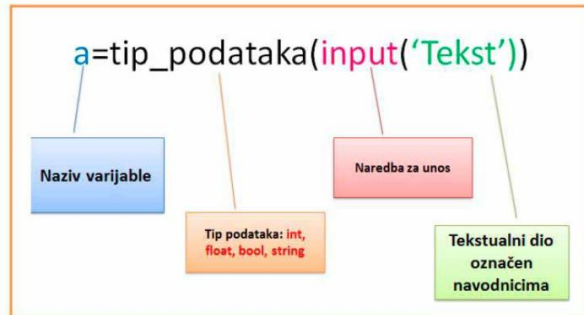
- a) Ime varijable smije sadržavati samo brojeve, velika i mala slova engleske abecede i donju crticu _
- b) Ime ne smije početi s brojem
- c) Python razlikuje velika i mala slova

Treba voditi brigu o znaku pridruživanja da se ne poistovjeti sa dvostruko jednako == što predstavlja operator usporedbe.

Osnovni tipovi operatora s kojima varijable mogu raditi su:

Aritmetički		Relacijski	
Znak	Naziv operacije	Znak	Naziv operacije
+	Zbrajanje	>	Veće
-	Oduzimanje	<	Manje
*	Množenje	>=	Veće i jednako
/	Dijeljenje	<=	Manje i jednako
%	Dijeljenje s ostatkom (modularno dijeljenje)	==	Jednako
**	Potenciranje	!=	Različito

Logički		Istovjetnosti	
Znak	Naziv operacije	Znak	Naziv operacije
And	Logički i	is	Istovjetan-vraća vrijednost TRUE
Or	Logički ili	is not	Nije istovjetan-vraća vrijednost FALSE
Not	Logički ne		



OPERATORI:

RELACIJSKI OPERATORI

Relacijski operator	
>	Veće
<	Manje
<=	Manje ili jednako
>=	Veće ili jednako
==	Jednako
!=	različito

```
n = int(input('Prvi broj:'))
m = int(input('Drugi broj:'))
if n>m:
    print('Veći je', n)
else:
    print('Veći je', m)
```

ARITMETIČKI OPERATORI

Aritmetičke operacije	
+	Zbrajanje
-	Oduzimanje
*	Množenje
/	Dijeljenje
**	Potenciranje
%	Dijeljenje s ostatkom

```
n = int(input('Unesi broj:'))
if n / 2==0:
    print('paran')
else:
    print('neparan')
```

Naredbe u Python-u

INPUT koristimo da bi mogli manipulirati podacima i nad njima vršiti različite operacije. Kako bi omogućili upisivanje tekstualne ili brojčane vrijednost u Python programski jezik, potrebno je upotrijebiti naredbu `input`.

Naredba `input` sama za sebe ne znači ništa. Nju moramo pridružiti nekoj varijabli kao običnu brojčanu ili tekstualnu vrijednost.

PRINT je naredba za ispisivanje vrijednosti. Naredba za unos aktivira se ključnom riječi `print`.

Dakle naredba ispisa omogućuje:

- ispis vrijednosti na standardni izlaz (zaslon monitora)
- višestruke vrijednosti odvojene zarezom
- `print` sam dodaje razmak između višestrukih vrijednosti

GRANANJE:

IF-ELSE ODLUKE

Odluke u Pythonu su realizirane u obliku grananja ili if-else uvjetovanja.

One omogućavaju slijednom ili linijskom programu odabir jedne od dvije mogućnosti odnosno kretanje u jednom od odabranih smjerova izvršavanja naredbi.

<ul style="list-style-type: none">• ako je uvjet onda<ul style="list-style-type: none">• Naredba 1• Naredba 2• inače<ul style="list-style-type: none">• Naredba 1• Naredba 2 <p>Govorni jezik</p>	<p>if uvjet:</p> <p>Naredba1 Naredba2</p> <p>else:</p> <p>Naredba1 Naredba2</p> <p>Python</p>
---	--

Odluke u programiranju možemo prevesti u govorni jezik kao frazu: ako je (uvjet) onda.....inače. Ukoliko je uvjet ispunjen izvrši prvi blok naredbi u suprotnom prijeđi na blok naredbi nakon naredbe inače.

U Pythonu naredba za odluke koristi se kao ključna riječ IF (uvjet)ELSE.... Dakle ukoliko je uvjet ispunjen, izvršava se blok naredbi odmah ispod naredbe if, a ukoliko uvjet nije ispunjen, izvršava se blok naredbi ispod naredbe else.

ELIF ODLUKE

Ukoliko postoji potreba za ispitivanjem višestrukih uvjeta, standardna struktura odluke if-else ne zadovoljava. Stoga standardno if-else uvjetovanje nadopunjujemo dodatnom strukturom elif. Uvjeta može biti vrlo mnogo (N), onoliko koliko sam programer odredi. Svaki elif ispituje svoj uvjet i ako je uvjet ispunjen, izvršava se blok naredbi smješten u tijelu elif-a. Elif uvjetovanje možemo zamisliti kao uzastopno postavljanje if uvjeta

<pre>if uvjet: Blok naredbi Blok naredbi elif uvjet_1: Blok naredbi Blok naredbi elif uvjet_2: Blok naredbi Blok naredbi elif uvjet_N: Blok naredbi Blok naredbi else: Blok naredbi Blok naredbi</pre>	<pre>ako uvjet: Blok naredbi Blok naredbi inače-ako uvjet_1: Blok naredbi Blok naredbi inače-ako uvjet_2: Blok naredbi Blok naredbi inače-ako uvjet_N: Blok naredbi Blok naredbi inače: Blok naredbi Blok naredbi</pre>
--	---

U govornom jeziku ovu elif konstrukciju bi čitali ovako: Dakle ukoliko je prvi uvjet ispunjen izvrši blok naredbi inače prijeđi na ispitivanje sljedećeg uvjeta; ako je uvjet ispunjen, izvrši blok naredbi inače prijeđi na ispitivanje sljedećeg uvjeta. Zato smo konstrukciju elif preveli kao inače-ako jer ona i jest kombinacija odluke if (ako) i else (inače).

PROGRAMSKA PETLJA FOR

Programske petlje su programske strukture koji omogućavaju višestruko ponavljanje određenog dijela programskog koda.

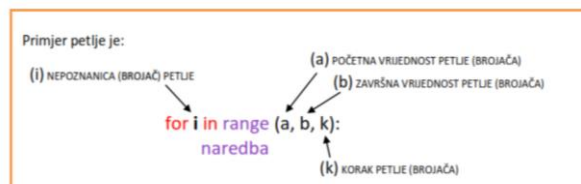
Petlja kao i programski konstrukt grananja ili odluke (if-else) sadržavaju uvjet.

Ispunjenjem uvjeta određeni dio koda, izvršava se zadani broj puta.

Osim toga petlja sadržava i brojač.

Brojač je nešto poput uvjeta, gdje zadajemo broj ponavljanja nekog programskog koda.

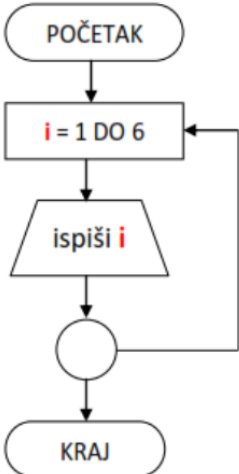
Ključna riječ za aktivaciju petlje je naredba for.



Uz nju dolazi oznaka i koja označava brojač. Brojač petlje zadaje broj ponavljanja određenog koda. Druga ključna riječ u petlji je range pomoću koje određujemo uvjet ponavljanja određenog koda. Unutar range nalazi se početna vrijednost brojača od koje petlja kreće ponavljati određeni kod, zatim završna vrijednost petlje te korak petlje kojim kažemo brojaču u kojem smjeru treba brojati i ponavljati petlju. Primjerice može se zadati ispis svakog drugog broja kojeg je korisnik unio. Ili se primjerice može zadati ispis negativnog broja. Završetak deklaracije označavamo sa znakom „:“ kao kod programskog konstrukta odluke.

Primjer programa:

Napišite program kojim ćete koristeći FOR petlju ispisati vrijednost brojača od 1 do 5.

DIJAGRAM TIJEKA	OBJAŠNJENJE
 <pre>graph TD; A([POČETAK]) --> B[i = 1 DO 6]; B --> C[/ispiši i/]; C --> D(()); D --> B; D --> E([KRAJ]);</pre>	Određujemo početnu i završnu vrijednost brojača.
	Ispiši svaku trenutačnu vrijednost brojača.
	Ponavljaj radnju dok dođeš do kraja brojača.