Programiranje

**Programiranje** je proces izrade programa.

**Program** je uputa računalu kako riješiti neki problem.

**Programski jezik** sadrži skup naredbi. Tim naredbama programer piše upute računalu prema kojima će ono riješiti neki problem.

Svaki program napisan je u nekom programskom jeziku. Programski jezik kojim ćemo se mi služiti jest [**Python**](https://www.python.org/)**.**

**Varijabla** je simboličko ime za memorijsku lokaciju.

Računalo u memoriju pohranjuje sve podatke i rezultate obrade. Budući da imena memorijskih lokacija nisu lako pamtljiva, u programiranju uvodimo simbolička imena za njih, tj. varijable.

**Imenovanje varijabli**

Varijable imenujemo sami, a preporučljivo ih je imenovati tako da imena označavaju ono što u njih pohranjujemo. U Pythonu, kao i u većini programskih jezika, vrijede sljedeća pravila:

* ime varijable smije sadržavati samo brojeve, velika i mala slova abecede te crticu
* ime varijable ne smije započeti brojem.
* Python razlikuje velika imala slova

**Svaka varijabla ima:**

* Ime
* Vrijednost
* Adresu memorijske lokacije

U varijablama pohranjujemo: brojeve, slova i sve druge znakove.

**SINTAKSA PROGRAMSKOG JEZIKA:**

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:

Slika na kojoj se prikazuje tekst, snimka zaslona, softver, Ikona na računalu

Opis je automatski generiran

Slika na kojoj se prikazuje tekst

Opis je automatski generiran

Primjer imenovanja varijable

**OPERATORI:**

Slika na kojoj se prikazuje tekst

Opis je automatski generiran

Slika na kojoj se prikazuje tekst

Opis je automatski generiran

**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.

Slika na kojoj se prikazuje tekst

Opis je automatski generiran

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

Slika na kojoj se prikazuje tekst, snimka zaslona, softver, Ikona na računalu

Opis je automatski generiranSlika na kojoj se prikazuje tekst, snimka zaslona, monitor, računalo

Opis je automatski generiran

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.

Slika na kojoj se prikazuje tekst

Opis je automatski generiran

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.

Slika na kojoj se prikazuje tekst, snimka zaslona, softver, Ikona na računalu

Opis je automatski generiran