FORMATIRANI ISPIS ;kao printf u C-u

'%[duljina][.preciznost]<tip>' % <varijable>

MODULI (.py)

```
import <imeModula>          ;unosi i inicijalizira sve elemente iz modula
<imeModula>.<atribut>       ;pristup elementima
dir(<imeModula>)            ;sva imena iz modula
reload(<imeModula>)         ;forsirani reload modula
from <imeModula> import <popisAtributa ili *>     ;unosi atribute
```

ZAPIS BROJEVA

HEX – **0x**<broj>
OKT – **0**<broj>

PRETVORBA U TIP

```
int(<source>)    hex(<source>)    oct(<source>)
str(<source>)    dict(<source>)
list(<source>)   tuple(<source>)
zip(<sekvence>)        ;lista n-torki elemenata sekvenci
```

USPOREDBE

```
<objekt> == <objekt>    ;sud, uspoređuje jednakost objekata
<objekt> is <objekt>    ;sud, uspoređuje adrese u memoriji
```

TOK PROGRAMA

;blokovi definirani jednakom indentacijom ;zagrade opcionalne
;u petljama **break**, **continue** i **pass**, opcionalni **else** se izvodi ako nije bilo **break**-a

FOR ;efektivno foreach

;ako je objekt niz n-torki radna varijabla može biti n-torka radnih varijabli kojima se može pristupati zasebno

```
for <rednaVarijabla> in <objekt>:
 <blokNaredbi>
else:
 <blokNaredbi>
```

WHILE

```
while <uvijet>:
 <blokNaredbi>
else:
 <blokNaredbi>
```

IF

```
if <uvijet>:
 <blokNaredbi>
elif <uvijet>:
 <blokNaredbi>
else:
 <blokNaredbi>
while <uvijet>:
 <blokNaredbi>
else:
 <blokNaredbi>
```

FUNKCIJE ;**return** opcionalan, default objekt **None** ;argumenti bez tipova

```
;posebi argumenti - *<ime>   ;non-keyword arguments – daje n-torku
                  - **<ime> ;keyword arg – daje hash
<drugoIme> = <imeFunkcije> ;mjenjamo ime objekta funkcije, argumenti ostaju isti
def <ime>(<argumenti>):
   <blokNaredbi>
   return <varijabla ili vrijednost>
```

LAMBDA IZRAZI

;slično funkciji koja ima samo jedan red bloka naredbi ;može biti anonimna

```
lambda <argumenti>: <izraz>
```

KLASE ;**self** obavezno eksplicitno kao prvi argument svake funkcije u klasi

```
class <ime>(<opcionalno – klase koje nasljeđuje>):
   def __init__(self, <argumenti>):          ;konstruktor
  <blokNaredbiKonstruktora>
```

TIPOVI

RJEČNIK (HASH) {...} ;mutable

```
<hash>[<key>]                          ;indexiranje ključem ;vraća vrijednost
<hash>.has_key (<key>)                 ;sud prisutnosti
<ključ> in <hash>                      ;isto kao prethodno
<hash>.keys()                          ;lista svih ključeva
<hash>.values()                        ;lista svih vrijednosti
<hash>.copy()                          ;nova kopiju hasha
<hash>.get(<key>, <default>)           ; vrijednost za ključ ili default ako nije nađen
len (<hash>)                           ;broj ključeva hasha
<hash>[<key>] = <value>                ;dodavanje para
del <hash>[<key>]                      ;brisanje para prema ključu
<hash>.items()                         ;lista n-torki (ključ, vrijednost)
<hashTarget>.update(<hashSource>)      ;dodaje sve parove iz izvornog u ciljni uz prepisivanje postojećih
<hash>.pop(<key>)                      ;briše iz hasha i vraća par prema ključu
```

TIPOVI

LISTA [...] ;mutable

```
<lista> [<indeks>]                                  ;0-indexirani pristup ;negativni od kraja
<lista> [<početniIndex>:<završniIndex+1>]           ;podlista ;mogu se izostaviti indexi
len(<lista>)                                        ;broj elemenata (duljina) liste
<lista> + <lista>                                   ;ulančavanje lista
<lista> * <broj>                                    ;umnažanje
<lista>.append(<element>)                           ;dodaje na kraj liste
<lista>.extend(<elementi>)                          ;dodaje na kraj liste
<lista>.sort()                                      ;sortiranje liste
<lista>.index(<element>)                            ;vraća indeks elementa ;error ako ga nema
<lista>.reverse()                                   ;invertira listu
del <lista>[<indeks>]                               ;briše element na indexu
del <lista>[<početniIndex>:<završniIndex+1>]        ;briše elemente na indeksima
<lista>.pop (<index>)                               ;briše iz liste i vraća indexirani element ;indeks default -1
<lista>[<početniIndex>:<završniIndex+1>] = []       ;isto kao del ;bez indexa briše sve
for <element> in <lista>                            ;foreach petlja
<element> in <lista>                                ;sud prisutnosti
range(<završniIndex+1>)                             ;podlista od 0-tog elementa
range(<početniIndex>, <završniIndex+1>)             ;podlista
range(<početniIndex>, <završniIndex+1>, <korakIndexa>)   ;podlista svakog <korak> indexa u intervalu
```

N-TORKA (TUPLE) (...) ;nonmutable

```
(<element>, )                                       ;jednočlana n-torka
<tuple>[<index>]                                    ;0-indexirani pristup
<tuple>[<početniIndex>:<završniIndex+1>]            ;pod-n-torka ; mogu se izostaviti indexi
len(<tuple>)                                        ;duljina (broj elemenata) n-torke
<tuple> + <tuple>                                   ;ulančavanje
<tuple> * <broj>                                    ;umnažanje
for <element> in <tuple>                            ;foreach petlja
<element> in <tuple>                                ;sud prisutnosti
```

DATOTEKA (FILE) ;zapis mora biti eksplicitno pretvoren u/iz texta

```
<file> = open (<path&name>, <način>)    ;otvara file ;način rada 'r' čitanje ili 'w' pisanje
<file>.read()                           ;string-cijeli file
<file>.read(<brojByteova>)              ;string-učitani Byteovi
<file>.readline()                       ;string-linija iz filea
<file>.readlines()                      ;lista stringova-linija filea
<file>.close()                          ;zatvara file ;nije nužno
```

STRING '...' ;nonmutable

prefix '**r**' ili '**R**' ispred navodnika – sirovi string-nema interpretacije escape znakova

```
<string> = """..."""                                ;kroz više redaka
<string> + <string>                                 ;konkatenacija ;implicitno i bez '+'
<string> * <broj>                                   ;umnažanje
<string> [<index>]                                  ;0-indexirani pristup znaku u nizu ;negativni indexi od kraja
<string> [<početniIndex>:<završniIndex+1>]          ;substring ;mogu se izostaviti indexi
len(<string>)                                       ;broj elemenata (duljina) niza
<string>.split(<razdjelnik>)                        ;daje listu razdjeljivanjem na razdjelniku ;implicitno su razdjelnici praznine
<string>.replace(<old>, <new>, <brojZamjena>)       ;ako se broj ne navede, rade se sve zamjene
<string>.find (<text>)                              ;vraća index prvog pojavljivanja ili -1 ako nema
<glue>.join(<lista>)                                ;daje string iz liste konkatenacijom glue-a između elemenata
<string>.rstrip()                                   ;uklanja '\n' sa kraja stringa
for <char> in <string>                              ;foreach petlja
<traženiString> in <stringUKomTražimo>              ;sud prisutnosti
eval(<string>)                                      ;vraća prepoznate podatkovne tipove iz stringa
```