

Коментари

Сваки коментар започиње симболом #, који се назива знаком бројева. Било шта после тог знака до краја исте линије кода представља коментар.

Када напише компјутерски програм, своје идеје програмер преточи у велики број линија кода. Програмерске идеје често нису очигледне било коме другоме сем самом аутору. Дешава се да аутор кода после неког времена се врати написаном коду али ни њему самом више није лако да одреди смисао његових идеја.

Једини начин да се избегну овакви проблеми је писање коментара. Коментари у коду помажу читању кода, олакшавају праћење развоја програмерских идеја и дају смисао и иначе тешко разумљивим линијама кода програмских језика.

Пример 01 Употреба коментара у коду

```
print('Naslov: Novi kod') #komentar zauzima deo linije koda
#komentar zauzima celu liniju bez koda
```

Излаз:

```
Naslov: Novi kod
```

Промењиве

Програми обично смештају податке у меморију рачунара и изводе операције над тим подацима. Програми користе промењиве за приступ и манипулацију подацима смештеним у меморији.

Промењива је име које представља вредност у компјутерској меморији.

Каже се да промењива упућује (reference) на вредност.

Идентификатор и оператор доделе

```
a = 23
```

Промењива **a** указује на меморијску локацију у којој се налази вредност **23**.

То је постигнуто помоћу **оператора доделе** (=, знак једнакости) (assignment operator).

Промењиве се представљају помоћу **идентификатора**.

```
a = 23
```

```
b = 11
```

```
name = "GOOD"
```

Промењива **a** указује на меморијску локацију у којој се налази вредност **23**, промењива **b** указује на локацију са вредности **11**, промењива **name** указује на локацију са вредности **GOOD**.

Свака промењива може да промени вредност током рада програма.

Промењива садржи њој прво додељену вредност све док се промењивој не додели нека друга вредност. Прво додељена вредност више није у употреби па Пајтон интерпретер аутоматски је брише из меморије преко процеса који се зове сакупљање ђубрета (*garbage collection*), чиме се ослобађа меморијска локација за даље коришћење.

Када се промењивој додељује вредност коју већ има нека друга промењива, интерпретер ће упутити да обе промењиве показују на исту меморијску локацију са жељеном вредности.

На овај начин се штеди употреба меморијских локација.

Провера указивања на исту локацију

Постоје два начина за проверу да ли промењиве указују на исту меморијску локацију:

Први је коришћењем инструкције **id(a)** унутар функције **print**, чиме се на екрану исписује адреса меморијске локације на коју упућује промењива **a**.

Други је коришћењем логичког оператора **is**.

Оператор упоређује адресе меморијских локација на које упућују промењиве **a** и **b** и ако су адресе идентичне, даје као резултат тачно (**True**) а у супротном нетачно (**False**).

Правила давања имена промењивима(избор идентификатора)

- не може се користити службена реч
- у имену не може постојати празно место
- први карактер може бити било које велико или мало слово или доња линија () (*underscore*)
- следећи карактери у имену могу бити слова, цифре или доње линије
- постоји разлика између малих и великих слова (*ja* није иста промењива као *Ja* или *jA*)
- име промењиве би требало да има описно значење о сврси промењиве (*tezina_u_uncama, baza_podataka_ucenici, tabela_repertoar_bioskopa*)