Promenljive

Promenljive se mogu posmatrati kao kontejneri koji sadrže podatke u sebi. Svaka promenljiva mora imati svoje ime. Promenljive su odličan način za skladištenje informacija i omogućavaju lako pristupanje tim informacijama kasnije u kodu. Prosta analogija za promenljive može biti kutija sa imenom koja u sebi sadrži određeni podatak. U Pythonu, kako je sve objekat – tako su i promenljive samo imena koja se daju objektima kako bi se oni identifikovali.

Deklarisanje promenljivih

Postoje dve faze u deklarisanju promenljivih:

1. Promenljiva se inicijalizuje tako što se kreira kontejnerski objekat kome će pripadati i dodeljuje joj se ime.
2. Dodeljuje joj se vrednost koristeći neki od operatora pridruživanja.

Pošto je Python dinamički tipiziran programski jezik, programer nije u obavezi da pri inicijalizaciji deklariše i tip te promenljive, ali je to dobra praksa.

Napomena

Inicijalizacija promenljive znači definisati joj ime i dodati inicijalnu, početnu vrednost toj promenljivoj. Promenljive koje nisu inicijalizovane su promenljive koje nemaju tu početnu vrednost, već im je samo ime definisano.

Pravila imenovanja promenljivih

Postoje određena pravila kojih programer treba da se pridržava prilikom imenovanja promenljivih:

- promenljive mogu početi samo slovima, velikim ili malim, od A do z (A–Z/a–z) ili donjom crtom (_);
- ostatak imena promenljive može sadržati slova, brojeve kao i donju crtu;
- Python razlikuje mala i velika slova, pa tako promenljive pod imenom „Test" i „test" nisu iste;
- ime promenljive ne može biti isto kao jedna od rezervisanih reči koje se mogu naći u sledećoj tabeli:

Rezervisane reči u Pythonu					
and	def	False	import	not	True
as	del	finally	in	or	try
assert	elif	for	is	pass	while
break	else	from	lambda	print	With
class	except	global	None	raise	Yield
continue	exec	if	nonlocal	return	

Tabela 7.1. Rezervisane reči u Pythonu

Nekoliko primera ispravnog i neispravnog imenovanja promenljivih

Ispravan način	Neispravan način	Razlog
moj_broj	moj-broj	Crte nisu dozvoljene u imenu promenljive.
broj1	1broj	Ne može početi brojem.
moj_broj	$moj)broj	Simboli u imenima nisu dozvoljeni.
moj_broj	moj broj	Ime promenljive mora sadržati jednu reč.

Tabela 7.2. Primeri ispravnih/neispravnih načina za imenovanje promenljivih

Dodeljivanje vrednosti

U Pythonu, dodeljivanje se vrši tako što se napiše prvo ime promenljive ispraćeno operatorom dodele (uglavnom (=)) i na kraju vrednost: ime_promenljive = "vrednost"

U sledećem odlomku koda vidimo primere za dodelu brojeva i stringova promenljivim:

Primer dodeljivanja vrednosti

Radno okruženje

```
year = 2020
print(year)

pi_value = 3.14
print(pi_value)

programming_language = "python"
print(programming_language)
```

Objašnjenje:

Dodajmo celobrojnu vrednost promenljivi year, promenljiva pi_value će imati vrednost decimalnog tipa a string dodajemo promenljivoj programming_language.

Višestruka dodela

U jednom izrazu možemo dodeliti više vrednosti za više promenljivih. To radimo na sledeći način:

Primer višestruke dodele

```
name, age ='Marco', 24
print(name, age)
```

Objašnjenje:

Promenljiva name će imati vrednost 'Marco' a age 24, tako da će ispis biti Marco 24. Ako u izrazu iz prvog reda ne dodamo drugu promenljivu, već samo napišemo `name = 'Marco', 24`, dobićemo tuple() tip podatka. Ako ne želimo tuple() tip podatka, ali i ne želimo da čuvamo obe vrednosti već samo jednu, možemo proslediti (_) karakter na mesto vrednosti koje ne želimo, pa bi prethodni izraz, ukoliko ne bismo želeli broj godina, izgledao ovako:

```
name, _ = 'Marco', 24
```

Ovim se u promenljivi _ čuva vrednost 24 i ako napišete print(_) dobićemo 24 ali pravilo nalaže da ne bismo trebali da je koristimo u programu.

Dodela iste vrednosti ka više promenljivih vrši se na sledeći način:

```
name = age = "Marco"
```

Zamena promenljivih

Zamena promenljivih se zapravo odnosi na menjanje njihovih vrednosti. Da bismo zamenili Python promenljive, koristimo sledeću logiku:

Primer zamene Python promenljivih

```
a, b=1, 2
print(a, b)
a, b=b, a
print(a, b)
```

Prvi ispis će biti 1,2 a kada promenljive a i b zamene vrednosti ispis će biti: 2,1.

Promenljive u Pythonu, tačnije ta imena, pokazuju samo na memorijsku adresu na kojoj se ta vrednost nalazi. Ako inicijalizujemo jednu promenljivu da pokazuje na vrednost druge i odmah potom promenimo vrednost druge promenljive na drugu vrednost, vrednost prve promenljive ostaje netaknuta, što vidimo u sledećem primeru:

Primer

```
a = 5
b = a
print(a, b)
a = 6
print(a, b)
```

Pitanje

Označiti netačnu tvrdnju:

- Ime promenljive može početi karakterima (_).
- Ime promenljive sme sadržati karakter A–Z/a–z.
- **Ime promenljive sme sadržati broj na početku imena promenljive.**

Objašnjenje:

Ime promenljive sme početi jedino karakterima A–Z/a–z ili _, dok se brojevi (cifre) smeju pojaviti u imenu promenljive tek nakon prvog znaka.

Brisanje promenljivih

Brisanje promenljivih se vrši pomoću ključne reči del:

Primer

```
a = 5
del a
```

Probajte da ispišete prethodni kod u radnom okruženju:

Radno okruženje

Probajte da ispišete promeljivu a posle njenog brisanja. Zašto dolazi do greške?

Greška se javlja jer program pokušava da pristupi promenljivi koja više ne postoji jer je prethodno izbrisana, isto bi se desilo i da pristupa promenljivi koja nikad nije ni napravljenja.

Rezime

- Promenljive nam služe za skladištenje informacija koje želimo da koristimo kasnije u kodu.
- Svaka promenljiva mora imati svoje ime.
- Imena promenljivih mogu početi samo slovima (A–Z/a–z) ili donjom crtom (_).
- Ostatak imena promenljive može sadržati slova, brojeve, kao i donju crtu.
- Programski jezik Python razlikuje mala i velika slova.
- Ime promenljive ne može biti isto kao jedna od rezervisanih reči koje se mogu naći u tabeli 7.1.
- Imenovanje promenljive i dodela vrednosti promenljivoj nazivaju se inicijalizacija.
- Dodela vrednosti se vrši pomoću operatora dodele.
- Python podržava višestruku dodelu vrednosti odjednom.
- Brisanje promenljivih se vrši ključnom rečju del.