

UPIS PODATAKA S TIPKOVNICE

U Pythonu, kao i u brojnim drugim programskim jezicima, znak jednakosti (=) je operator pridruživanja. Ako želimo pohraniti broj 5 na nepoznanici **a**, onda jednostavno pišemo:

```
a = 5
```

Naredba **input()** omogućuje da se upisani broj ili tekst zapamti na varijabli.
Primjer:

```
print('Upiši prvi broj: ')\n\na = input()\n\nprint('Upiši drugi broj: ')\n\nb = input()\n\nzbroj = int(a) + int(b)\n\nprint('Zbroj je', zbroj)
```

Ovaj program možemo pisati i kraće:

```
a = int(input('Upiši prvi broj: '))\n\nb = int(input('Upiši drugi broj: '))\n\nprint('Zbroj je', a + b)
```

Naredba **int()** pretvara upisani podatak u cijeli broj.

Pravila za imenovanje varijabli:

1. ime varijable se sastoji od slova, znamenki i podvlake (_).
2. Python razlikuje velika i mala slova tako da npr. **Rezultat** i **rezultat** nisu iste varijable.
3. ime varijable ne može započeti znamenkom
4. ime varijable ne može biti ključna riječ u Pythonu (npr. *if*, *or*...)

KAKO DONOSIMO ODLUKE

Budući da se računala temelje na binarnom sustavu (dva stanja, 0 i 1), program može donijeti odluku o tome koje će se naredbe izvršiti ako mu postavimo logički uvjet (0=laž, 1=istina). Npr, ako u Python shell upišemo:

```
3>2
```

Python će odgovoriti:

```
True
```

Operatori za uspoređivanje u Pythonu su:

- < (manje)
- > (veće)
- == (jednako)
- <= (manje ili jednako)
- >= (veće ili jednako)
- != (različito)

Primjer 1: Program provjerava jesu li dva broja jednaka

```
a = int(input('a='))
```

```
b = int(input('b='))
```

```
if a == b:

    print('Brojevi su jednaki.')

else:

    print('Brojevi su različiti.')
```

Primjer 2: Program provjerava je li neki broj djeljiv s drugim brojem

```
a = int(input('a='))

b = int(input('b='))

if a % b == 0:

    print(a, 'je djeljiv s', b)
```

Postoji i ključna riječ **elif** (skraćeno od *else if*) koja se koristi kad imamo više uvjeta.

KAKO PONAVLJATI NAREDBE

Naredba:

```
for k in range (n):

    print(k)
```

zove se **programska petlja** ili jednostavno petlja. Ona omogućuje ponavljanje naredbi zadani broj puta. Varijablu **k** zovemo **varijabla petlje** ona mijenja vrijednost pri svakom prolasku petljom. Ako želimo brojeve ispisati jedan pored drugoga, razdvojene razmakom, pišemo

```
print(k, end=' ')
```

Raspon (*range*) u **for** petlji možemo zadati na više načina:

- `for k in range(10)` – varijabla **k** počinje s 0 i završava s brojem 9. Pri svakom prolazu petljom poveća se za 1.
- `for k in range(1,11)` – varijabla **k** počinje s 1 i završava s brojem 10. Pri svakom prolazu petljom poveća se za 1.
- `for k in range(10,1,-2)` – varijabla **k** počinje s 10 i završava s 1. Pri svakom prolasku petljom smanji se za 2.

PETLJA S LOGIČKIM UVJETOM

Ako trebamo u programiranju koristiti petlju, a ne znamo koliko puta će se naredbe unutar petlje ponavljati, koristit ćemo petlju `while`.

Primjer: Kod igre “Čovječe ne ljuti se” potrebno je dobiti šesticu za početak igre. Napiši program koji će brojiti iz kojeg pokušaja je dobivena šestica.

```
x = pokusaja = 0

while x != 6:

    x = input('Upiši broj: ')

    x = int(x)

    pokusaja += 1

print('Dobio si šesticu iz',pokusaja,'. pokušaja.')
```

Petlja započinje ključnom riječi `while` iza koje slijedi **logički uvjet** i dvotočka (:). Naredbe unutar petlje izvršavaju se sve dok je logički uvjet na početku petlje istinit. Ako je logički uvjet pokraj `while` lažan prije nego što se petlja počne izvoditi, program će preskočiti cijelu petlju.