**Odločitve (nadaljevanje)**

**Več o pogojih**

Do sedaj smo za pogoje pisali samo logične izraze, lahko pa bi pogoj bil kakršen koli izraz. Pravimo, da pogoj **ni** izpolnjen, če ima eno izmed naslednjih vrednosti: False, 0, 0.0, "", '', (), [], {}, set(), None. V vseh ostalih primerih je pogoj izpolnjen.

Ko Python vrednoti pogoj, ki vsebuje logične operatorje and in/ali or, ga vrednoti samo tako daleč, da lahko določi vrednost celotnega pogoja. V pogoju p1 and p2 and p3 bi tako najprej ovrednotil pogoj p1, če je ta izpolnjen, še pogoj p2, če je tudi ta izpolnjen, pa še pogoj p3. Če kateri od teh pogojev ne bi bil izpolnjen, mu preostanka ni treba vrednotiti, saj vemo, da celoten pogoj ne more biti izpolnjen. V takem primeru vrne vrednost pogoja, ki ni izpolnjen. Podobno velja za pogoj p1 or p2 or p3.

Primerjave lahko tudi verižimo: x < y < z pomeni isto kot x < y and y < z

**Krajše oblike prireditvenega stavka**

Pri programiranju bomo večkrat morali spremeniti vrednost izbrane spremenljivke, tako da ji bomo nekaj prišteli, jo pomnožili, ... Dolga oblika stavka, ki vrednost spremenljivke a poveča za 1, bi bila a = a + 1, s čimer stari vrednosti spremenljivke a prištejemo 1 in dobljeno vrednost shranimo nazaj v a. Krajše bi to zapisali a += 1 in prebrali: vrednost spremenljivke a povečaj za 1. Podobne okrajšave pozna Python tudi za druge aritmetične operatorje:

|  |  |
| --- | --- |
| a = a + 3  | a += 3  |
| a = a - 3  | a -= 3  |
| a = a \* 3  | a \*= 3  |
| a = a \*\* 3  | a \*\*= 3  |
| a = a / 3  | a /= 3  |
| a = a // 3  | a //= 3  |
| a = a % 3  | a %= 3  |

**Primeri**

1. Sestavi funkcijo, ki v danem štirimestnem naravnem številu prešteje, koliko števk je večjih od k.
2. Sestavi funkcijo, ki dano kompleksno število izpiše čim lepše. Kompleksno število podamo z realno in imaginarno komponento.