

TINGIMUSLAUSE IF

Kõik eelnevalt käsitletud programmid olid lineaarse struktuuriga: kõik käsud täideti järjestikku ükshaaval, iga kirjutatud käsk täidetakse tingimata. Oletame, et tahame antud arvu x jaoks määrata selle absoluutväärtuse (mooduli). Programm peaks printima muutuja x väärtuse, kui $x > 0$, või väärtuse $-x$ vastasel juhul. Programmi lineaarne struktuur on rikutud: olenevalt tingimuse $x > 0$ kehtivusest tuleb väljastada üks või teine väärtus.

Tingimuslauses (ka valikulause või if-lause) on kesksel kohal tingimus, mille alusel otsustatakse, kuidas programm edasi töötab. Tingimus on esitatud tõeväärtuse tüüpi avaldisena, mille väärtus on tõene (True) või väär (False) – tingimus on täidetud või ei ole täidetud.

Tingimusi saab (muuhulgas) moodustada järgmiste märkidega:

- $<$, $>$, $<=$, $>=$ võrratuste kehtivuse kontrollimiseks;
- $=$ võrdsuse kontrollimiseks;
- $!=$ mittevõrdsuse kontrollimiseks.

Tingimuslausel võib olla else-osa, millega määratakse, mis juhtub siis, kui tingimus ei ole täidetud (vastava avaldise väärtus on False).

```
In [ ]: x = 10
        if x > 0:
            print(x)
        else:
            print(-x)
```

10

elif Kui on vaja sisestada mitu alternatiivset tingimust, saab kasutada täiendavaid elif-plokke, millele järgneb käsuplokk.

Esiteks kontrollib Python if-lauset. Kui see on True (Tõene), täidetakse if-plokis olevad laused. Kui see tingimus tagastab väärtuse False (Väär), kontrollib Python avaldist elifist.

...

Kui avaldis pärast elif on True, siis täidetakse elif-plokis olevad käsud. Aga kui see on False, siis täidetakse käsud else-plokist.

Vajadusel saab erinevate tingimuste jaoks määratleda mitu elif-plokki.

```
In [ ]: language = "german"
if language == "english":
    print("Hello")
elif language == "german":
    print("Hallo")
elif language == "french":
    print("Salut")
elif language == "estonian":
    print("Tere")
else:
    print("-----")
```

Hallo

Pesastatud if konstruktsioonid

If-konstruktsioonil endal võivad olla pesastatud if-konstruktsioonid:

```
In [ ]: language = "english"
daytime = "morning"
if language == "english":
    print("English")
    if daytime == "morning":
        print("Good morning")
    else:
        print("Good evening")
```

English
Good morning

Siin sisaldab if-konstruktsioon pesastatud if/else konstruktsiooni. See tähendab, et kui muutuja language on võrdne "english", siis pesastatud if/else konstruktsioon kontrollib täiendavalt daytime muutuja väärtust, kas see on võrdne sõnega "morning" või mitte.

Peab arvestama, et pesastatud if-avaldised peavad algama samuti taandega ja pesastatud konstruktsioonides olevad käsud samuti taandega. Valesti paigutatud taane võib muuta programmi loogikat.

```
In [ ]: language = "english"
daytime = "morning"
if language == "english":
    print("English")
if daytime == "morning":
    print("Good morning")
else:
    print("Good evening")
```

English
Good morning

Ülesanne 1

Saime sama tulemuse, kuid skriptide täitmine on erinev. Kontrollige, mida skriptid väljastavad, kui language = "estonian".

Loogilised tehted

Loogilisi tehteid kasutatakse liitingimusavaldiste loomiseks. Pythonil on järgmised loogilised operaatorid:

and (loogiline korrutamine)

Tagastab väärtuse True, kui mõlemad avaldised on True.

```
In [ ]: age = 28
        height = 168
        if age > 20 and height < 180:
            print('True')
```

True

Sel juhul võrdleb operaator and kahe avaldise tulemusi: age > 21 height < 180. Ja kui mõlemad avaldised tagastavad väärtuse True, tagastab ka and operaator väärtuse True.

or (loogiline liitmine)

Tagastab True, kui vähemalt üks avaldistest on True

```
In [ ]: month = 3

        if month == 3 or month == 4 or month == 5:
            print("Spring")
```

Spring

not (loogiline eitus)

Tagastab väärtuse True, kui avaldis on False

```
In [ ]: if not month==3:
        ...
```

Kui üks and operaatori operant tagastab False, siis teist operandi enam ei hinnata, kuna operaator tagastab niikuinii väärtuse False. See käitumine võimaldab jõudlust veidi suurendada, kuna ei pea kulutama ressursse teise operandi hindamiseks.

Samamoodi, kui operaatori või üks operandidest tagastab väärtuse True, siis teist operandi ei hinnata, kuna operaator tagastab niikuinii väärtuse True.

Operaator `in`

`in` operaator tagastab True, kui mõnes väärtuste komplektis on teatud väärtus. Sellel on järgmine vorm:¶

```
In [ ]: month = 3

if month in [3, 4, 5]:
    print("Spring")
```

Spring

Ülesanne 2

Mis on järgmiste avaldiste True/False tulemus (ilma programmeerimiseta):

```
x=15; y=20
x>=10 and x<100
x==y or x==15
x==1 or x==2
y-x>0 and y<=20
x==15 and x==16 ✓
```

Algajate programmeerijate tavaline viga on loogiliste operaatorite and ja or segi ajamine. Käsitleme kahte tingimust:

if $x > 1$ and $x < 100$:

if $x > 1$ or $x < 100$:

Ainult esimene tingimus on tõene. See kontrollib, et arv x on vahemikus 1 kuni 100, teisisõnu $x \in (1;100)$.

Teine tingimus kontrollib, et arv x on kas suurem kui 1 või väiksem kui 100. Selle tingimuse täidab aga iga arv!

Pythonis saab võrdlusoperaatoreid ühendada **ahelateks** (erinevalt enamikust teistest programmeerimiskeeltest, kus selleks tuleb kasutada loogilisi sidemeid),

näiteks $a == b == c$ или $1 <= x <= 10$

```
In [ ]: a, b, c = 1, 1, 1

if a==b==c:
    print(f'a={a}, b={b}, c={c} are equals')
```

a=1, b=1, c=1 are equals

Näide

On antud kolm täisarvu. Määrake, kui palju neist langevad kokku. Programm peaks väljastama ühe arvudest: 3 (kui kõik langevad kokku), 2 (kui kaks langeb kokku) või 0 (kui kõik arvud on erinevad). 3 skripti täidavad sama ülesannet.

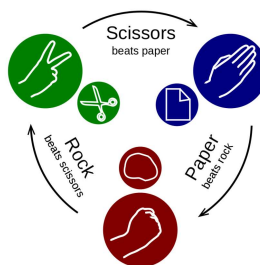
```
a, b, c = int(input()), int(input()), int(input())
if a == b:
    if b == c:
        print(3)
    else:
        print(2)
else:
    if a == c:
        print(2)
    else:
        if b == c:
            print(2)
        else:
            print(0)
```

```
a, b, c = int(input()), int(input()), int(input())
if a == b == c:
    print(3)
elif a == b != c:
    print(2)
elif a != b == c:
    print(2)
elif a == c != b:
    print(2)
else:
    print(0)
```

```
a, b, c = int(input()), int(input()), int(input())
if a == b == c:
    print(3)
elif a == b != c or a != b == c or a == c != b:
    print(2)
else:
    print(0)
```

Ülesanded

1. Kasutaja sisestab kaks numbrit ja matemaatilise operaatori (+, -, *, /) ning programm väljastab tulemuse. Kontrollige, et 0-ga jagada ei tohi.
2. Kasutaja sisestab kolmnurga kolm külge ja kontrollib, kas see on kolmnurk? (kolmnurk on võimalik ainult siis, kui mis tahes kahe külje summa on suurem kui kolmas külge).
3. Looge testprogramm korrutustabeli teadmiste kontrollimiseks. Looge juhuslike arvude abil 5 näidet. Õige vastuse korral kirjutatakse ekraanile Õige, vale korral Viga. Samuti peab programm sõltuvalt testi sooritamise tulemusest panema hinde: kui õigete vastuste arv = 5, siis Tubli; 3 või 4, siis Peaks veel õppima; 2 või vähem – On vaja kiiresti selgeks õppida korrutustabel. Vihje. Õigete vastuste arvu lugemiseks luuakse muutuja, mida õige vastuse korral suurendatakse 1 võrra.
4. Kirjutage kivi-paber-käärid mäng. Kasutaja mängib arvutiga. Kasutaja sisestab tekstiga märgi ja arvuti valib juhuslikult 3 märgi hulgast. Programm määrab võitja. Kuidas saada juhuslikku täisarvu?



Kuidas saada juhuslikku täisarvu?

Funktsioon `randint()` moodulist `random`, mis võtab kaks argumenti – täisarvuvahemiku piirid, millest valitakse suvaline arv. Näiteks peate saama täisarvu vahemikus 1 kuni 5. Näiteks peate saama täisarvu vahemikus 1 kuni 5.

```
In [ ]: import random

        rand_number = random.randint(1, 5)
        print(rand_number)
```

3

Märkus: Skripti moodulid ühendatakse üks kord ja parem on see kirjutada koodi algusesse.