

Mänguline Pythoni programmeerimine: Ülesanded

Sisend-väljund, muutujad ja aritmeetika

Ülesanne 1

Kirjuta enda esimene programm, mis väljastab käsureale teksti: "Tere, maailm!". Programm näeb välja selline:

```
print("Tere, maailm!")
```

Jooksuta seda programmi vajutades IDLE tekstiredaktoris klahvi F5.

Ülesanne 2

Küsi kasutajalt sisend tema vanuse kohta ning väljasta see ekraanile:

```
print("Kirjuta vanus:")  
vanus = input()  
print(vanus)
```

Ülesanne 3

Mis tüüpi on järgnevad muutujad:

- a) `vanus = 18`
- b) `eesnimi = "Jaak"`
- c) `pikkus = 16.5`
- d) `kas_käib_koolis = True`

Mis võimalus veel peale `True` oleks viimast muutujat väärtustada? Kuidas võiks nende muutujate väärtusi koodis kontrollida?

Ülesanne 4

Mida teevad järgmised tehted?

- a) `vanus = 18 + 1`
`print(vanus)`
- b) `vanus = vanus - 5`
`print(vanus)`
- c) `vanus = 18 * 0`
`print(vanus)`
- d) `vanus = (vanus + 1) / 5`
`print(vanus)`

Ülesanne 5

Mida teeb järgnev kood?

```
eesnimi = "Jaak"  
perenimi = "Vilo"  
nimi = eesnimi + perenimi  
print(nimi)
```

Ülesanne 6

Mida teeb järgnev kood? Kuidas saada korrektne tulemus?

```
vanus = 18  
eesnimi = "Jaak"  
print(eesnimi + vanus)
```

Ülesanne 7 [M]

Kirjuta enda mängu koodis laual olevate kommide arv muutujasse. Seejärel kuva muutujas olev kommide arv ekraanile kasutades `print()` käsku.

Ülesanne 8 [M]

Küsi kasutajalt sisendit, mitu kommi ta soovib laualt ära võtta. Eemalda soovitud kommide arv laual olevate kommide arvust ja kuva ekraanile, kui palju komme laual nüüd on.

Tingimuslaused

Ülesanne 9

Kirjuta järgnev kood ja proovi, kuidas kood töötab erinevate muutujate *a* ja *b* väärtuste korral.

```
a=2
b=3
if a>b:
    print("a>b")
elif a<b:
    print("a<b")
else:
    print("a=b")
```

Ülesanne 10

Mida väljastavad järgnevad koodid?

a) a=0

```
if a<1 and a>-1:
    print("a on 0")
else:
    print("a ei ole 0")
```

b) a=2

```
b=3
if a==0 or a==0:
    print("a*b=0")
else:
    print("a*b!=0")
```

Ülesanne 11

Proovi, kas järgnev kood töötab õigesti ja leiab arvu absoluutväärtust:

```
print("Palun sisesta mingi arv: ")
arv_tekstina = input()
arv = float(arv_tekstina)
if arv < 0:
    vastus = -arv
else:
    vastus = arv
print("Selle arvu absoluutväärtus on " + str(vastus))
```

Ülesanne 12 [M]

Täiendame mängu koodi selliselt, et:

1. Kui mängija soovib eemaldada vähem kui 1 või rohkem kui 3 kommi, siis kuvatakse seda keelav teade ja komme ei eemaldata.
2. Kui mängija soovib eemaldada 1 kuni 3 kommi, ent väljakul ei ole piisavalt komme, siis kuvatakse seda keelav teade ja komme ei eemaldata. Selline olukord juhtub, kui väljakule on jäänud näiteks 2 kommi ja soovitakse eemaldada 3.
3. Kui ülal nimetatud tingimused ei täitunud, ehk siis mängija soovib eemaldada täiesti valiidses arvu komme, siis eemaldame need ja kuvame järelejäänud kommide arvu ekraanile.

Tsükkel

Ülesanne 13

Mida teeb järgnev kood?

```
arv = 1
while arv <= 100:
    print(arv)
    arv = arv + 1
```

Ülesanne 14

Kas järgnev kood töötab õigesti ja küsib kasutajalt sisendit seni, kuni see on 2?

```
print("Sisesta number 1-3:")
arv = int(input())
while arv != 2:
    print("Proovi uuesti! Sisesta number 1-3:")
    arv = int(input())
print("Arvasid õigesti! Arv oli 2.")
```

Ülesanne 15

Proovi aga järgnevat koodi, kas su sõber arvaks arvu 5 ära?

```
print("Sisesta number 1-10:")
arv = int(input())
while arv != 5:
    if arv < 5:
        print("Liiga väike arv! Sisesta number 1-10:")
        arv = int(input())
    else:
        print("Liiga suur arv! Sisesta number 1-10:")
        arv = int(input())
print("Arvasid õigesti! Arv oli 5.")
```

Ülesanne 16 [M]

Täiendame mängu koodi selliselt, et kasutaja saaks eemaldada komme nii kaua kuni neid on.

Funktsioonid

Ülesanne 17 [M]

Lisame mängu selle, et kasutajalt küsitakse sisendit nii kaua kuni see saab vastama ülesandes 12 mainitud tingimustele. Ehk siis, kui kasutaja sisend ei olnud valideeritud, siis küsime kohe uut sisendit.

Ülesanne 18 [M]

Lisame mängu vastase käigu. Selleks loome uue muutuja *arvuti_käik*, mis alguses on väärtustatud tõeväärtusega *False* ning pärast mängija käiku saavutab väärtuse *True*. Pärast arvuti käiku muudame uuesti selle väärtuse *False*.

Vastavalt selle muutuja väärtusele küsime, kas kasutajalt sisendit või siis laseme arvutil eemaldada kommidest 1 kommi ning kuvame seda kirjeldava teate ekraanile.

Programm näeks välja umbes:

```
arvuti_käik = False
while kommid_arv > 0:
    if arvuti_käik:
        kommid_arv -= 1
        print("Arvuti võttis ühe kommi")
        arvuti_käik = False
    else:
        #Kasutaja sisendi lugemine muutujasse "mitu_kommi_võtta"
        kommid_arv -= mitu_kommi_võtta
        arvuti_käik = True
```

Ülesanne 19 [M]

Olukorras, kus laualt saavad kommid otsa ja läheme põhitsüklist välja (*kommid_arv > 0*), tuleb meil otsustada, kes siis mängu võitis. Mõelda välja, kuidas olemasolevaid andmeid kasutades seda lihtsasti kontrollida ja kuvada vastav teade ekraanile.

Ülesanne 20

Mida väljastab järgnev kood?

```
def väljasta(väärtus):  
    print(väärtus)  
  
a = 10  
väljasta(a)  
b = 11  
väljasta(b)
```

Ülesanne 21

Mida väljastab järgnev kood?

```
def liida(a, b):  
    return a - b  
  
a = 10  
b = 11  
print(liida(b, a))
```

Ülesanne 22 [M]

Tõsta kasutaja sisendi lugemine ja valiidsuse kontroll eraldi funktsiooni. See funktsioon saab sisendiks hetkel laual olevate kommide arvu, sest seda on valiidsuse kontrollis vaja.

Funktsioon tagastagu kasutaja poolt võetavate kommide arv. Funktsiooni tagastust saab kasutada mängija käigu tegemiseks.

Ülesanne 23

Milleks on Pythonis olemas erinevad moodulid? Kust nende kohta infot leida?

Ülesanne 24

Milliseid funktsioone sisaldab *random* moodul?

Mida teeb funktsioon *random.randint(a, b)*?

Ülesanne 25 [M]

Täiendame mängu nii, et arvuti võib võtta juhusliku arvu komme 1-st 3-ni. Kas sellega võib tekkida probleem mängu töös? Mis täpselt?

Ülesanne 26 [M]

Lahendame eelmises ülesandes vihjatud probleemi.

Ülesanne 27 [M]*

Millised oleksid võitvad strateegiad arvutile selle mängu mängimiseks? Proovi mõndasi nendest rakendada.