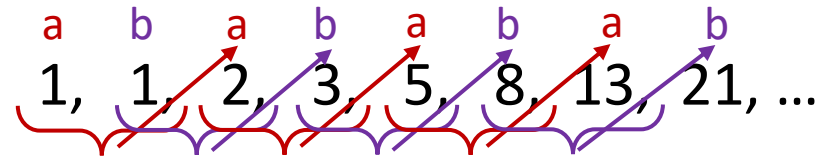


## t. 471, s. 130

Yksinkertaisin (lyhin) versio, jossa Fibonaccin luvut tulostetaan omille riveilleen:

Tulostetaan luvut silmukassa aina pareittain ( $a$  ja  $b$ ). Määritelmän mukaisesti lukujonon luvut saadaan aina kahden edellisen summana:



Talletetaan kahden edellisen jäsenen summa  $a + b$  ensin muuttujaan  $a$ , ja sitten muuttujaan  $b$ . Tässä järjestyksessä uutta  $b$ :n arvoa laskettaessa huomioidaan uusi  $a$ :n arvo.

```
a=1  
b=1
```

```
for i in range(5):  
    ♦♦ print(a)  
    ♦♦ print(b)  
    ♦♦ a=a+b  
    ♦♦ b=a+b
```

Silmukkaa toistetaan viisi kertaa (viiden parin verran).

Tuloste:

```
1  
1  
2  
3  
5  
8  
13  
21  
34  
55  
>>>
```

Tässä versiossa Fibonaccin luvut tulostetaan lukujonon muodossa yhdelle riville pilkulla erotettuna.

```
a=1
```

```
b=1
```

```
for i in range(5):  
    ♦♦ if i<4:  
        ♦♦♦♦ print(a, end=", ")  
        ♦♦♦♦ print(b, end=", ")  
    ♦♦ else:  
        ♦♦♦♦ print(a, end=", ")  
        ♦♦♦♦ print(b, end=", ...")  
    ♦♦ a=a+b  
    ♦♦ b=a+b
```

Oletuksena "print" -komento tekee rivinvaihdon tulosteen jälkeen. Parametrilla "end" voidaan määrittää välimerkiksi esimerkiksi pilkku (ja välilyönti) , jolloin tulostusta voidaan jatkaa samaan "listaan".

Silmukan sisällä on nyt ehtolause. Viimeisen tulostettavan luvun jälkeen tulostetaan ", ..."

Huomaa, että viimeinen i:n arvo on 4, eikä 5 (koska alkuarvo on 0).

Tuloste:

```
1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...
```

```
>>>
```

Tässä versiossa käytetään while-silmukkaa ja yhtä tulostusta jokaisella toistokerralla.

```
a = 1
b = 1
i = 3
#Kaksi ensimmäistä tulostetaan ennen toistoa käyttäen välimerkkinä pilkkua.
print(a, end = ", ")
print(b, end = ", ")
while i<=10:
    ♦♦if i<10:
# Tässä tulostetaan kahden edellisen summa käyttäen välimerkkinä edelleen pilkkua.
# Ensin tulostuu siis 1+1 = 2.
    ♦♦♦♦print(a + b, end = ", ")
    ♦♦else:
# Viimeinen eli kymmenes Fibonaccin luku tulostetaan ilman pilkkua.
    ♦♦♦♦print(a + b)
# Tallennetaan muistiin vanha a:n arvo.
    ♦♦muistiin = a
# Tallennetaan uudeksi a:n arvoksi Fibonaccin lukujonon seuraava luku eli b.
    ♦♦a = b
# Lasketaan seuraava b:n arvo lisäämällä vanhaan b:n arvoon alkuperäinen a:n arvo.
    ♦♦b = b + muistiin
# Kasvatetaan laskurin arvoa yhdellä.
    ♦♦i = i + 1
```