



Kul med kod

Micro:bit och Micropython

4

Övning 4. Lek med 25 lysdioder

```
from microbit import *
import random
```

while True:

```
    for x in range(0, 5):
        y = random.randint(0, 4)
        display.set_pixel(x, y, 9)
        sleep(100)
    display.clear()
```

```
from microbit import *
import random
```

while True:

```
    for x in range(0, 5):
        y = random.randint(0, 4)
        b = random.randint(1, 9)
        display.set_pixel(x, y, b)
        sleep(100)
    display.clear()
```

Slumpen bestämmer

Importera biblioteket **random** som inte finns med i microbit utan måste importeras separat.

y = random.randint(0, 4) betyder att y får ett slumpstal 0-4 varje gång raden körs. Hur många lysdioder kommer att tändas varje gång?

Experimentera med **sleep(100)** och placera den lite på olika ställen. Flytta också på **display.clear()**

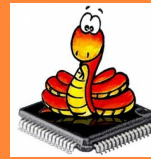
Blir resultatet som du förväntade dig?

Ta en titt på sista exemplet. Vilken skillnad ser du om du jämför första kodexemplet med det andra?

Vad tror du kommer att hända?

Innan du går vidare

Repetera och fundera kring de olika kodexemplen. Kommer du ihåg objektet display? Funktionen sleep()? Looparna? Variabler? Random osv...



Kul med kod

Micro:bit och Micropython

1

Övning 1. Lek med 25 lysdioder

```
from microbit import *
display.scroll("Hello World")
```

```
from microbit import *
```

while True:

```
    display.scroll("Hello World")
```

```
from microbit import *
```

while True:

```
    display.show("Hello World")
    display.show(Image.HEART)
```

```
from microbit import *
```

while True:

```
    display.show(Image.HEART)
    sleep(1000)
    display.clear()
    sleep(1000)
```

Koda först delen och skicka över till din micro:bit. Observera vad som händer!

Ändra koden som i nästa exempel. Vad händer nu?

Vad är skillnaden?

Observera indraget efter :

Testa nästa exempel och ladda upp det till kortet. Observera vad som händer.

Kan du stoppa in koden **sleep(1000)** någonstans för att se hur den påverkar det som visas. Fundera på varför.

Ladda upp igen och se skillnaden.

Läs och förstå sista exemplet, ladda upp koden och se resultatet.

Kan du få hjärtat att slå lite snabbare?

Vad tror du 1000 betyder i funktionen sleep(1000)?



Kul med kod

Micro:bit och Micropython

2

Övning 2. Lek med 25 lysdioder

```
from microbit import *
```

```
minbild = Image("09090:"
```

```
"09090:"
```

```
"00000:"
```

```
"90009:"
```

```
"09990")
```

```
display.show(minbild)
```

```
from microbit import *
```

```
<dina bilder här>
```

```
while True:
```

```
    display.show(minbild1)
```

```
    sleep(1000)
```

```
    display.show(minbild2)
```

```
    sleep(1000)
```

```
from microbit import *
```

```
x = 2
```

```
y = 2
```

```
display.set_pixel(x, y, 9)
```

Prova första koden.

Fungerar den?

Kan du utläsa "bilden" i koden?

Varje lysdiod kan kodas att lysa på olika sätt: inte alls (0) upp till starkt (9).

Din egen bild.

Skapa en egen bild med de 25 lysdioderna. Rita gärna på ett rutpapper med 5x5 rutor.

Utöka med en bild till.

Gör en liten animation med två bilder som du växlar mellan. Du behöver då göra två ex. minbild1 och minbild2

Du måste också använda en loop (while True:) för att se din animation.

Tänd en specifik lysdiod

Vi testar med två variabler x och y. Första lysdioden uppe till vänster har positionen x=0, y=0.

Kan du få lysdioden längst ned till höger att lysa svagt?

Tips: ändra 9 till ex. 4 eller 5.



Kul med kod

Micro:bit och Micropython

3

Övning 3. Lek med 25 lysdioder

```
from microbit import *
```

```
y = 2
```

```
for x in range(0, 5):
```

```
    display.set_pixel(x, y, 9)
```

```
from microbit import *
```

```
while True:
```

```
    for y in range(0, 5):
```

```
        for x in range(0, 5):
```

```
            display.set_pixel(x, y, 9)
```

```
            sleep(100)
```

```
from microbit import *
```

```
while True:
```

```
    for y in range(0, 5):
```

```
        for x in range(0, 5):
```

```
            display.set_pixel(x, y, 9)
```

```
            sleep(100)
```

```
    display.clear()
```

En loop fixar allt

Med loopen du använder kan du specificera hur många gånger den ska utföra sitt arbete. Observera indraget efter kolon. Nu har du lärt dig två olika loopar.

En loop nästlad i en loop

Här gäller det att hänga med, försök att läsa ut vad som kommer att ske.

Med **sleep(100)** gör du så att du hinner se vad som sker. Indraget gör så att **sleep** tillhör x-loopen.

Vad händer om du låter indraget (TAB) tillhöra y-loopen?

Vad händer om du tar bort **sleep(100)**?

Rensa displayen

Experimentera med funktionen **display.clear()**, prova olika positioner, dvs. indrag. Vad tror du kommer att hända?

Om **display.clear()** finns före **while True:** fyller den då någon funktion?