

# IELET1002 Datateknikk

## Øving 3 : Oppdrag 3 [millis() og RTH]

**Et Arduino reaksjonsspill**



**millis()**

## Oppdrag 3: Utforske millis()

1. Ta utgangspunkt i en kode du har fra før. Start programmet ditt med å lagre verdien du finner i fra millis(), altså det første som skjer i void loop. Dette er nå starttidspunktet til koden din. Det siste som skjer i loop, skal være å sjekke verdien til millis() igjen. Hvor mye større har millis blitt? Nå kan vi si hvor lang tid koden vår brukte på å kjøre.
2. Ta en titt på millis() i Språkmanualen (Language reference) på [arduino.cc](https://arduino.cc/language-reference), og forsøk å sette deg inn i alt som står der.
3. Gå nøye gjennom de to casene som følger i denne presentasjonen.
4. Og søk på nettet for å finne ut mer om hvordan millis() kan erstatte delay(), og mye mer enn det.

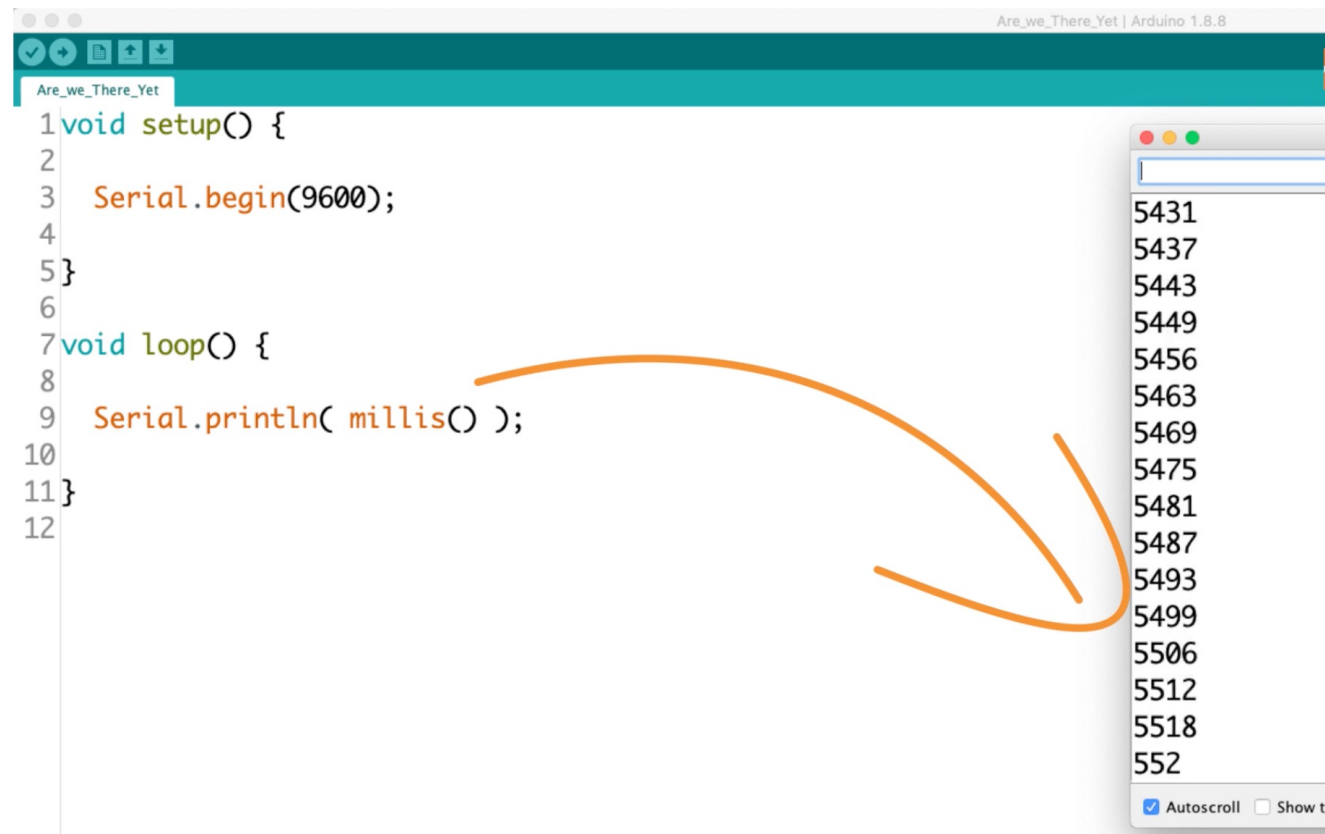
# Registrere tid:

*-Hvordan få koden til å gjøre noe  
etter at en viss tid har gått?*

# Hvor mye tid har gått?

millis()

*Teller antall  
millisekunder  
etter programstart*



The screenshot shows the Arduino IDE interface. The sketch editor on the left contains the following code:

```
1 void setup() {  
2  
3   Serial.begin(9600);  
4  
5 }  
6  
7 void loop() {  
8  
9   Serial.println( millis() );  
10  
11 }  
12
```

The serial monitor on the right displays the output of the sketch, showing a list of millisecond values: 5431, 5437, 5443, 5449, 5456, 5463, 5469, 5475, 5481, 5487, 5493, 5499, 5506, 5512, 5518, and 552. An orange arrow points from the `millis()` function call in the code to the serial monitor output.

Are\_we\_There\_Yet | Arduino 1.8.8

Are\_we\_There\_Yet

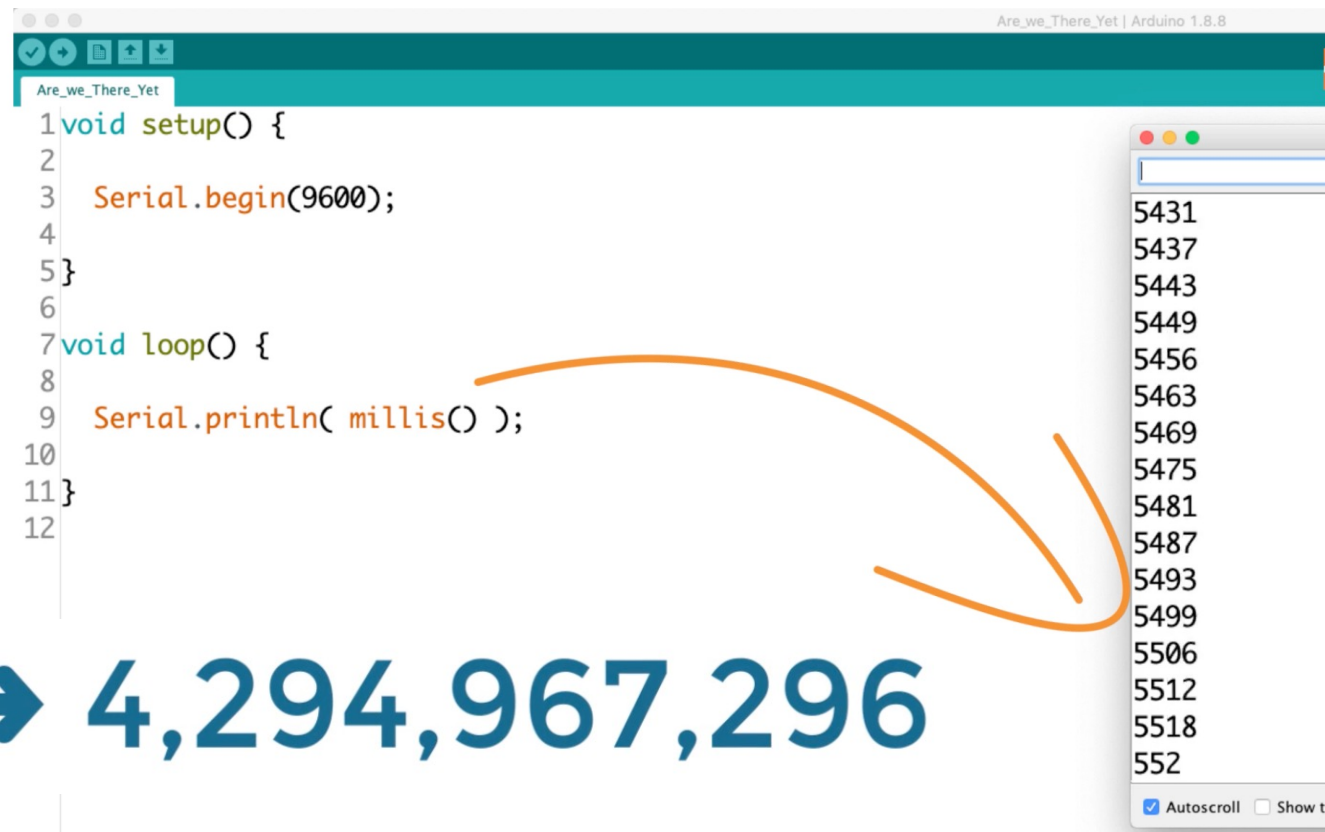
5431  
5437  
5443  
5449  
5456  
5463  
5469  
5475  
5481  
5487  
5493  
5499  
5506  
5512  
5518  
552

☒ Autoscroll ☐ Show ti

# Hvor mye tid har gått?

millis()

*Teller antall  
millisekunder  
etter programstart*



The screenshot shows the Arduino IDE interface. The sketch editor on the left contains the following code:

```
1 void setup() {  
2     Serial.begin(9600);  
3 }  
4  
5 void loop() {  
6     Serial.println( millis() );  
7 }  
8  
9  
10  
11  
12
```

The serial monitor on the right displays the output of the program, showing a list of millisecond values: 5431, 5437, 5443, 5449, 5456, 5463, 5469, 5475, 5481, 5487, 5493, 5499, 5506, 5512, 5518, 552. An orange arrow points from the `millis()` function call in the code to the serial monitor output.

0 → 4,294,967,296

# Hvor mye tid har gått?

**millis()**

*Teller antall  
millisekunder  
etter programstart*

Arduino Variable Sizes (for Arduino UNO)			
Data Type Name	Size in bits / bytes	From Size	To Size
byte	8-bits (1-byte)	0	255
integer (int)	16-bits (2-byte)	-32,768	32,767
float	32-bits (4-bytes)	-3.4028235E+38	3.4028235E+38
long	32-bits (4-bytes)	-2,147,483,648	2,147,483,647
unsigned long	32-bits (4-bytes)	0	4,294,967,295

Alle variabler og konstanter som benyttes i beregninger hvor **millis()** inngår MÅ deklarereres som unsigned long!

# Hvor mye tid har gått?



Vi ønsker å utføre noe 1000ms  
etter at programmet har startet

Hvordan sjekke om vi har kommet til et  
gitt “tidspunkt” etter oppstart av  
programmet?



# Hvor mye tid har gått?



# Hvor mye tid har gått?



# Hvor mye tid har gått?



# Hvor mye tid har gått?



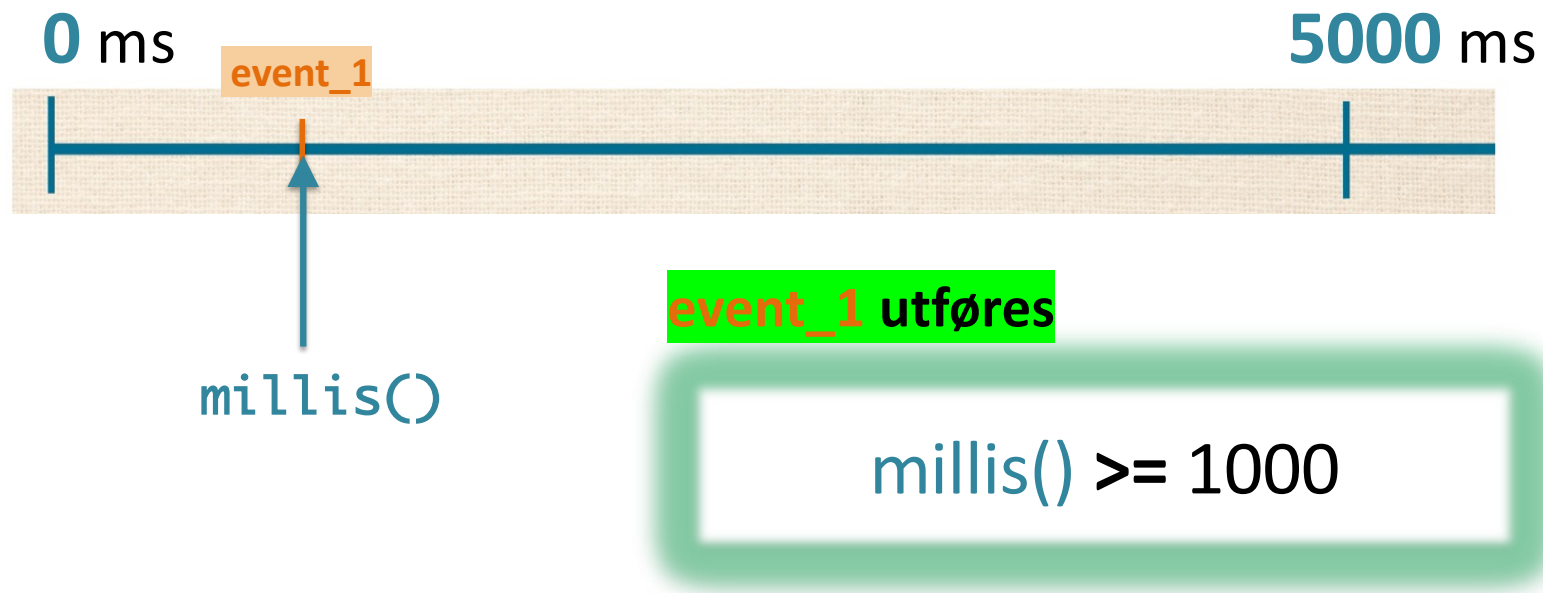
# Hvor mye tid har gått?



# Hvor mye tid har gått?



# Hvor mye tid har gått?



# Hvor mye tid har gått?

**millis()**

*Teller antall  
millisekunder  
etter starten av  
av programmet*

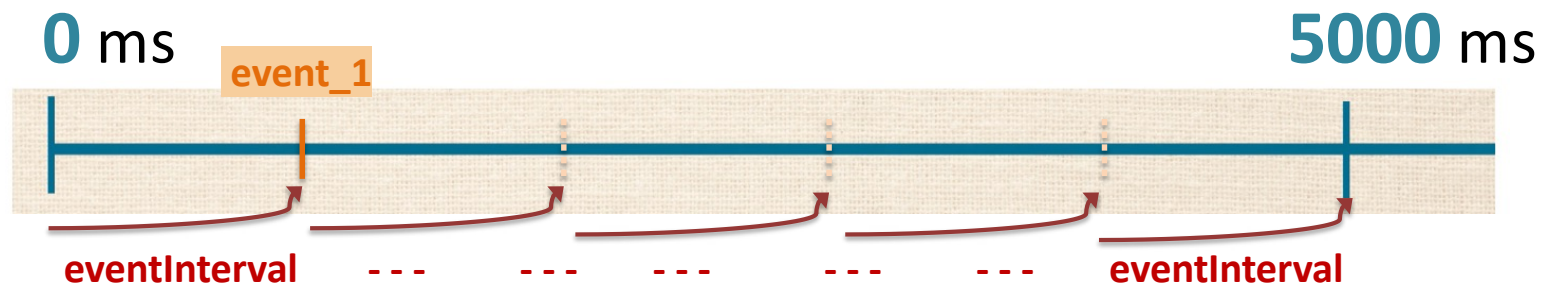
```
1 const unsigned long event_1 = 1000; // utfør event_1 etter 1000ms
2
3 void setup() {
4
5 }
6
7 void loop() {
8
9     if(millis() >= event_1) { // har millis() nådd 1000ms?
10
11         // utfør event_1 (bare hvis millis() har nådd 1000ms)
12
13     }
14 }
```



# Repeterende Timede Hendelser

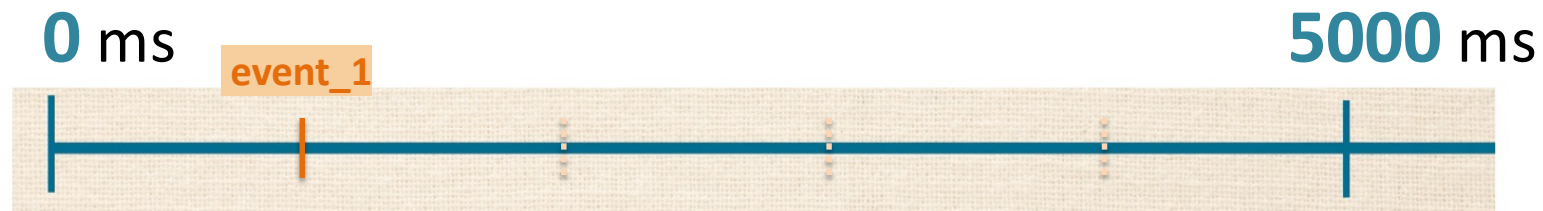
*-Hvordan få koden til å gjøre noe  
i faste tidsintervall?*

# Repeterende Timede Hendelser (RTH)



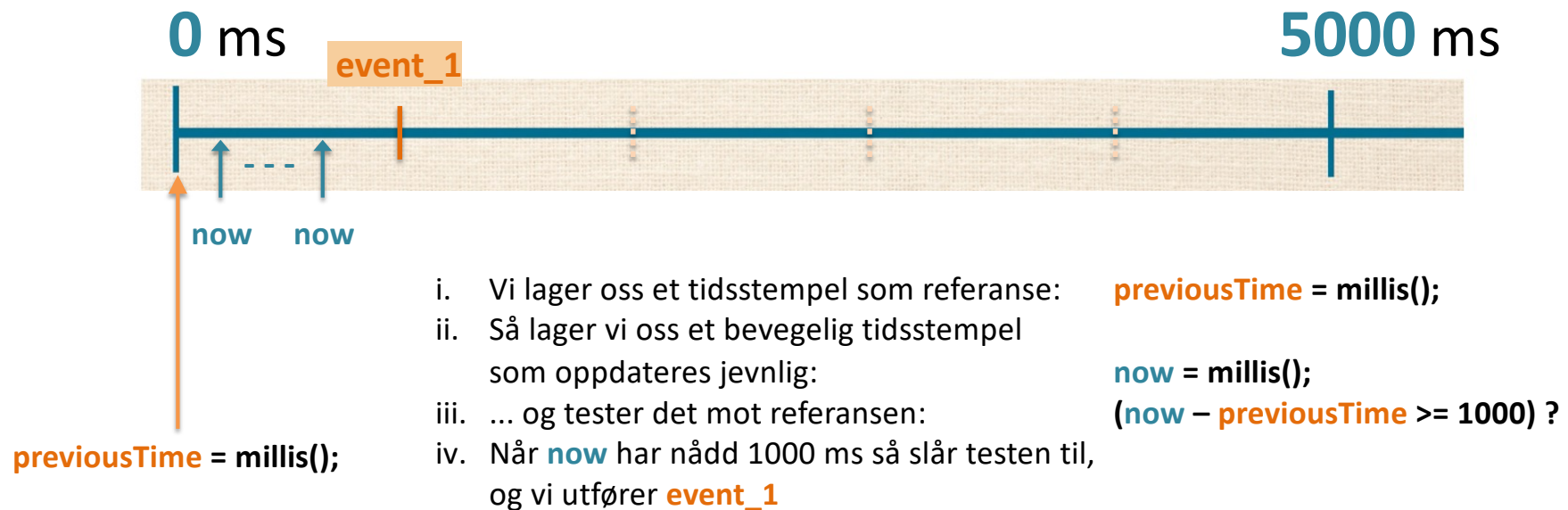
Hvordan gjøre noe i faste intervall?

# Repeterende Timede Hendelser (RTH)

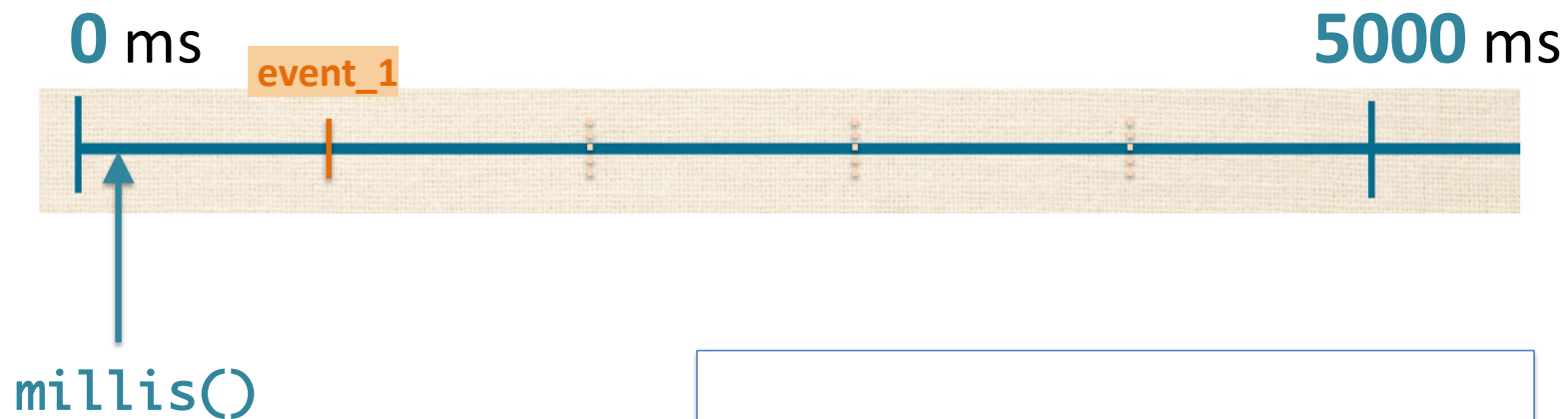


Hvordan gjøre noe i faste intervall?

# Repeterende Timede Hendelser (RTH)

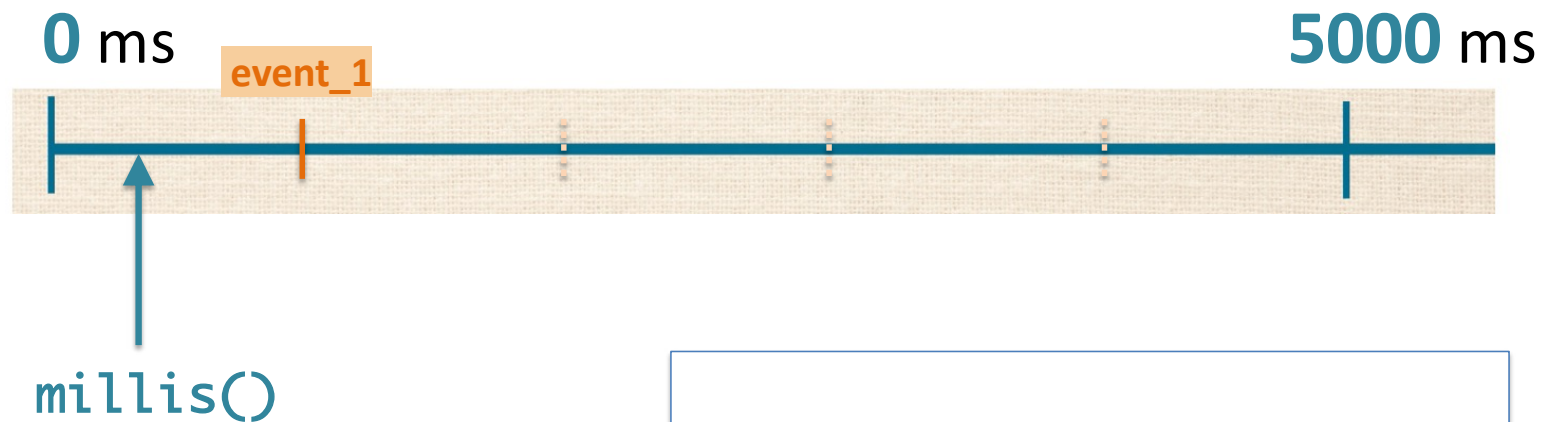


# Repeterende Timede Hendelser (RTH)



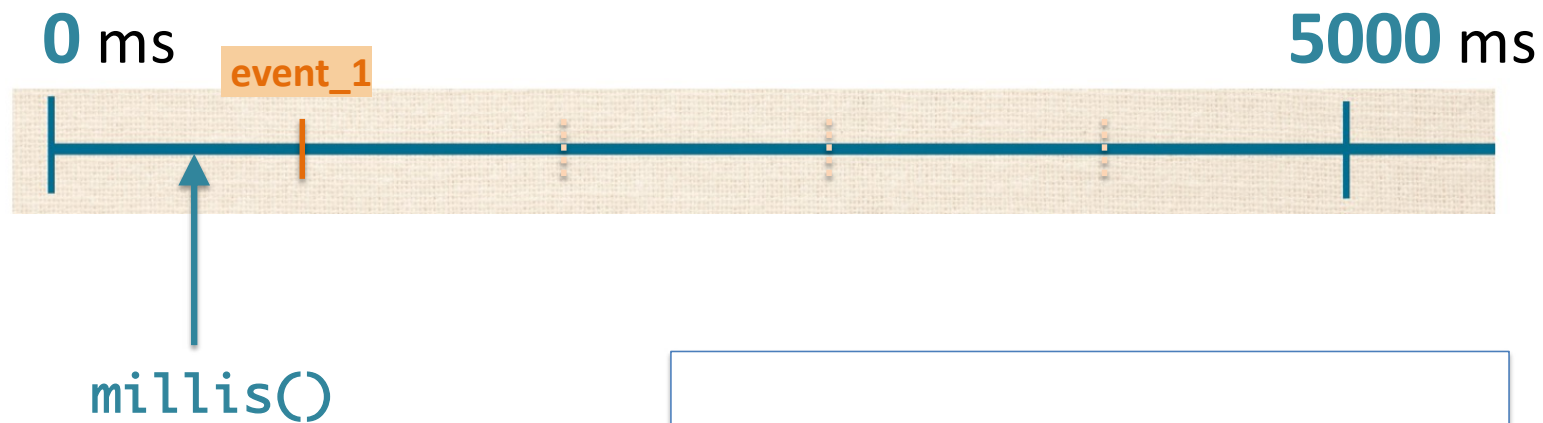
Hvordan gjøre noe i faste intervall?

# Repeterende Timede Hendelser (RTH)



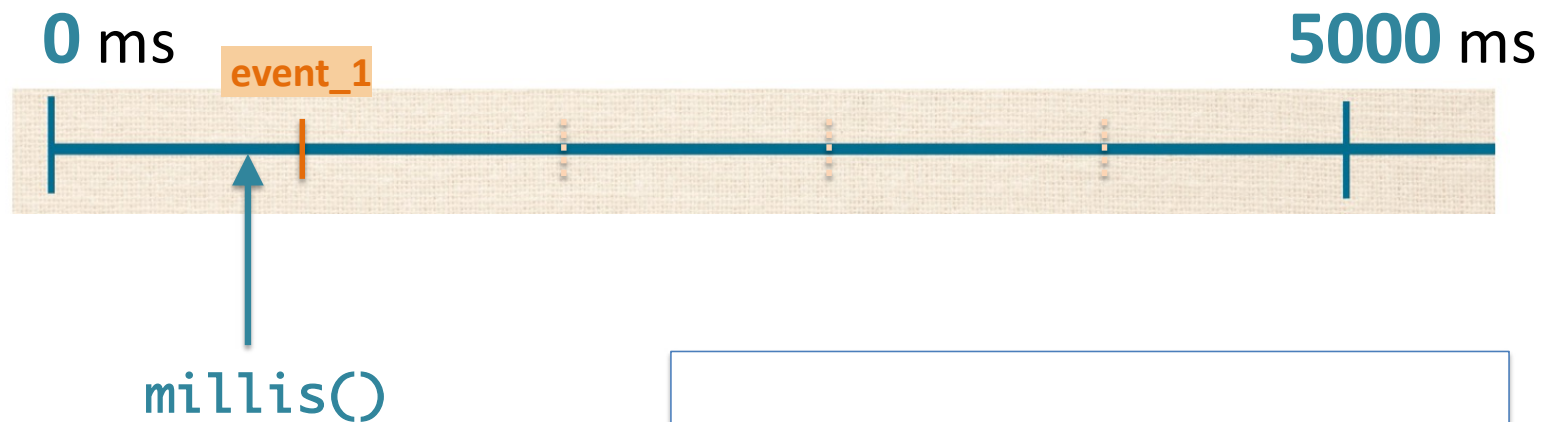
Hvordan gjøre noe i faste intervall?

# Repeterende Timede Hendelser (RTH)



Hvordan gjøre noe i faste intervall?

# Repeterende Timede Hendelser (RTH)



Hvordan gjøre noe i faste intervall?



# Repeterende Timede Hendelser (RTH)



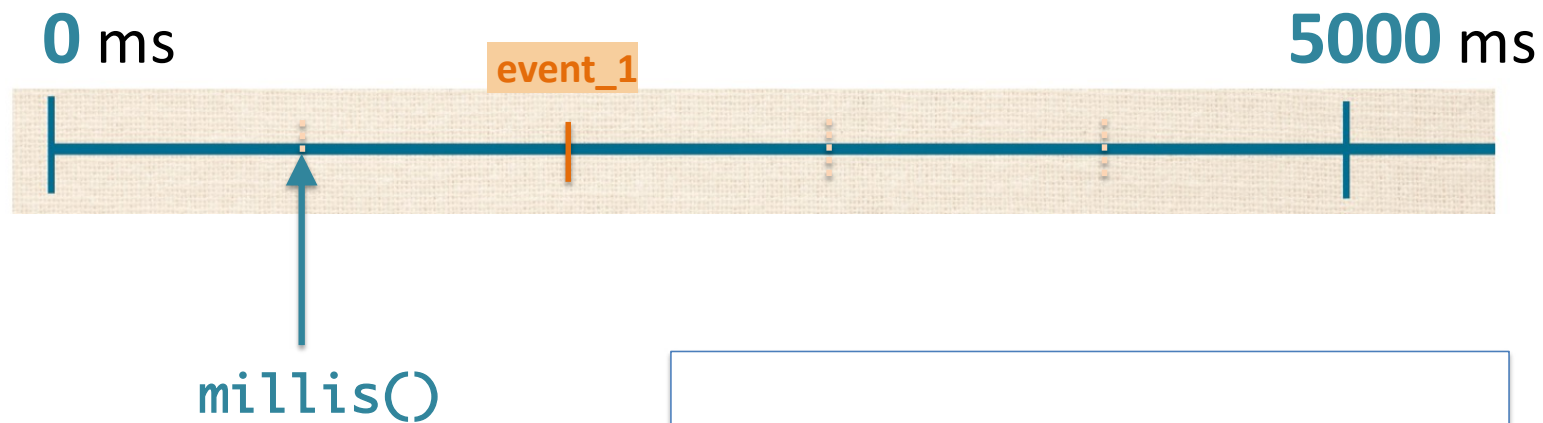
Da har `now = millis()` kommet fram til 1000ms (og `event_1`)

- testen slår til:  $(\text{now} - \text{previousTime} \geq 1000)$
- og `event_1` utføres

Men hva nå?

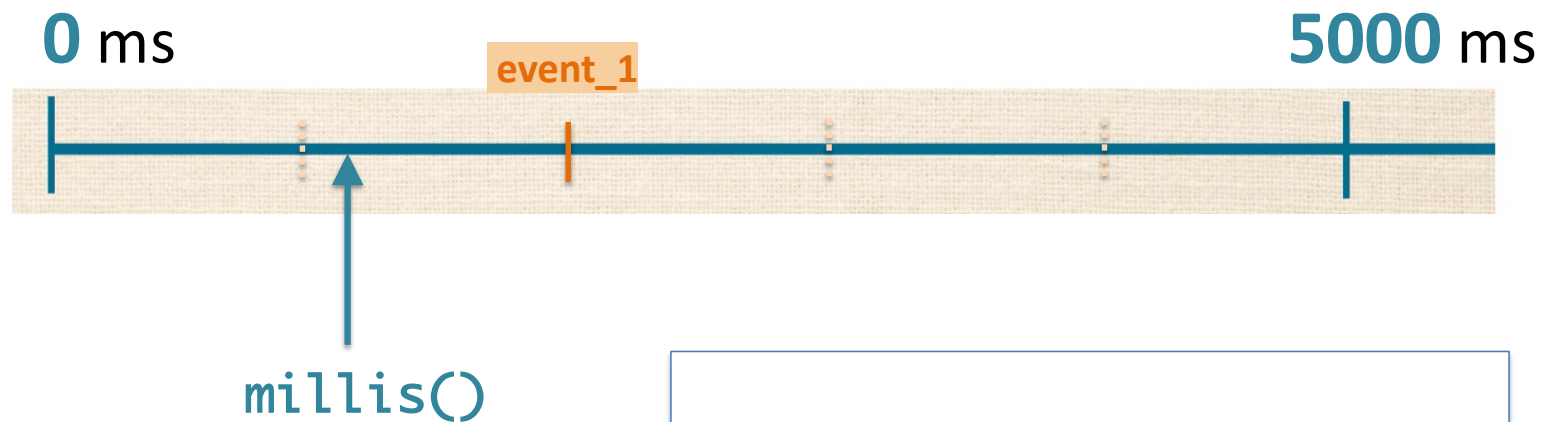
- Flytter fram referansen `previousTime` med 1000ms: `previousTime = now;`
- og så begynner vi å teste på stadig nye  $(\text{now} - \text{previousTime} \geq 1000)$
- dvs vi har også **flyttet fram `event_1` med nye 1000ms**

# Repeterende Timede Hendelser (RTH)



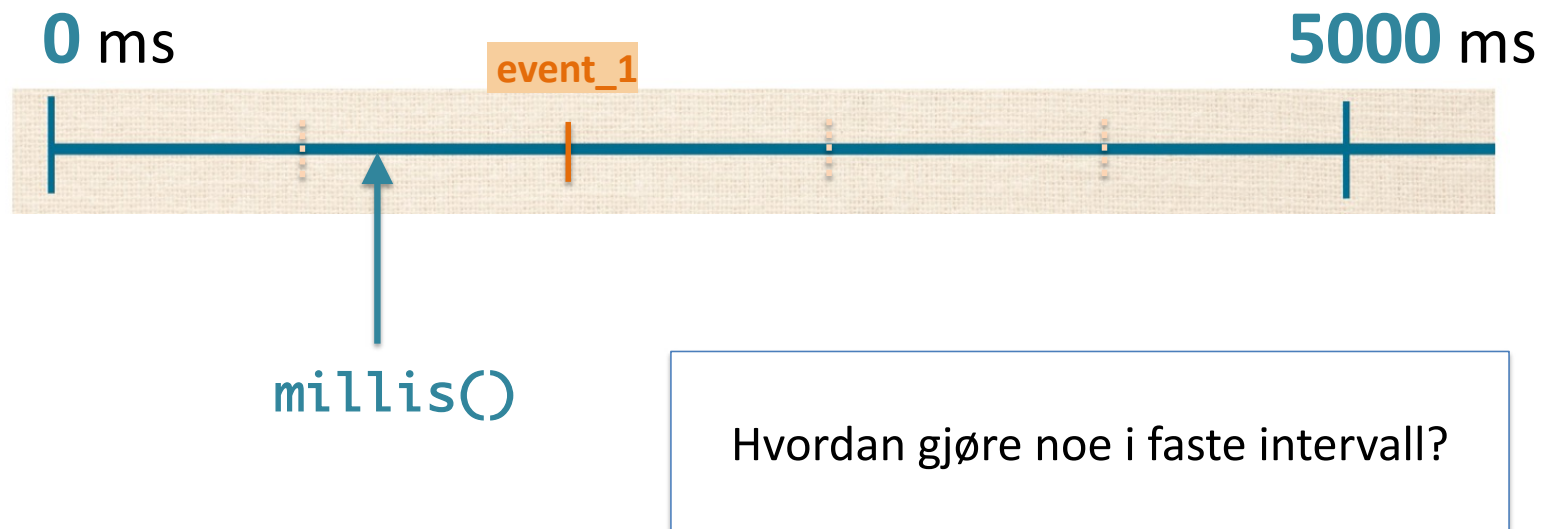
Hvordan gjøre noe i faste intervall?

# Repeterende Timede Hendelser (RTH)

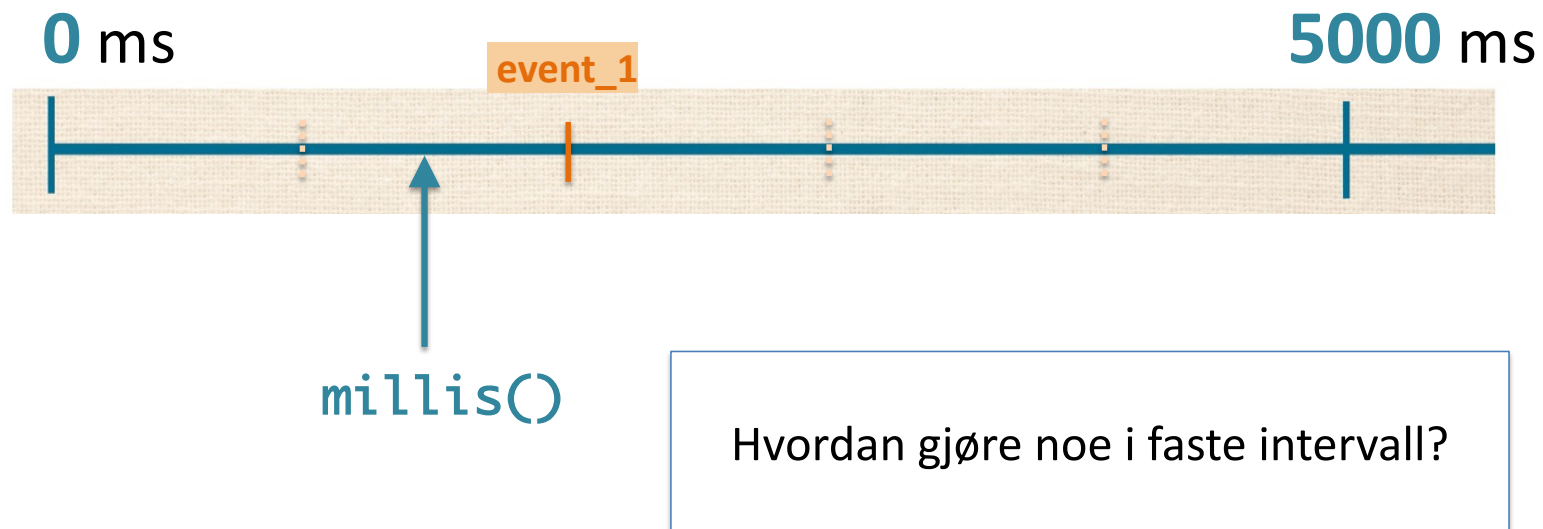


Hvordan gjøre noe i faste intervall?

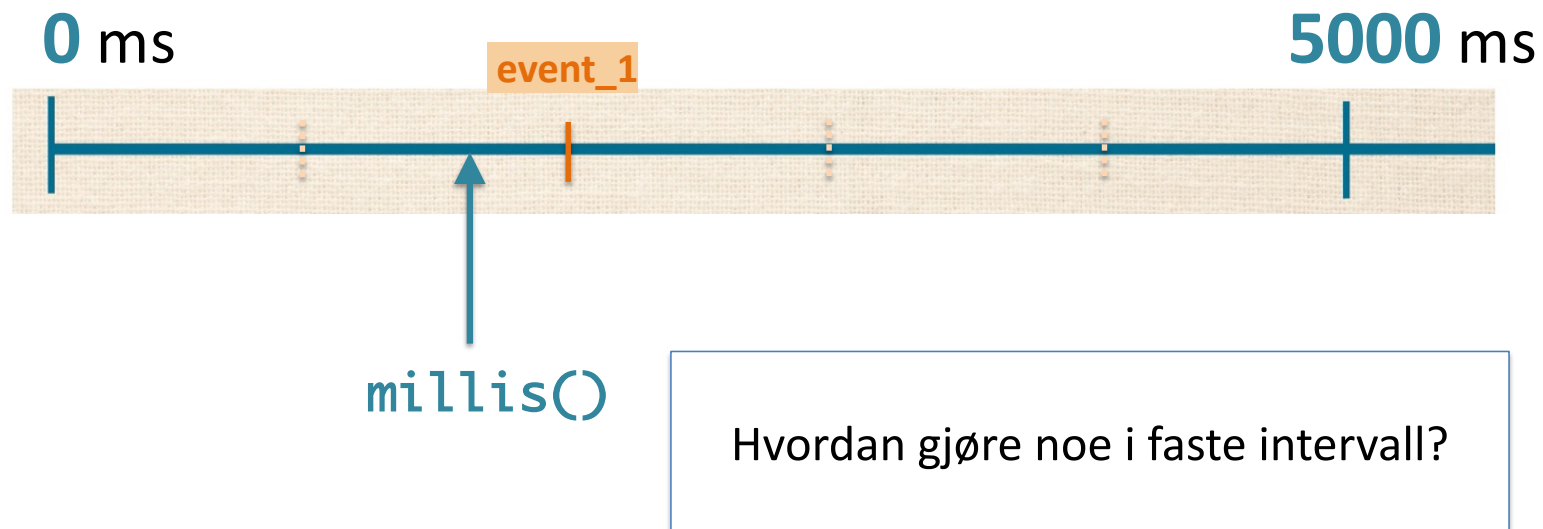
# Repeterende Timede Hendelser (RTH)



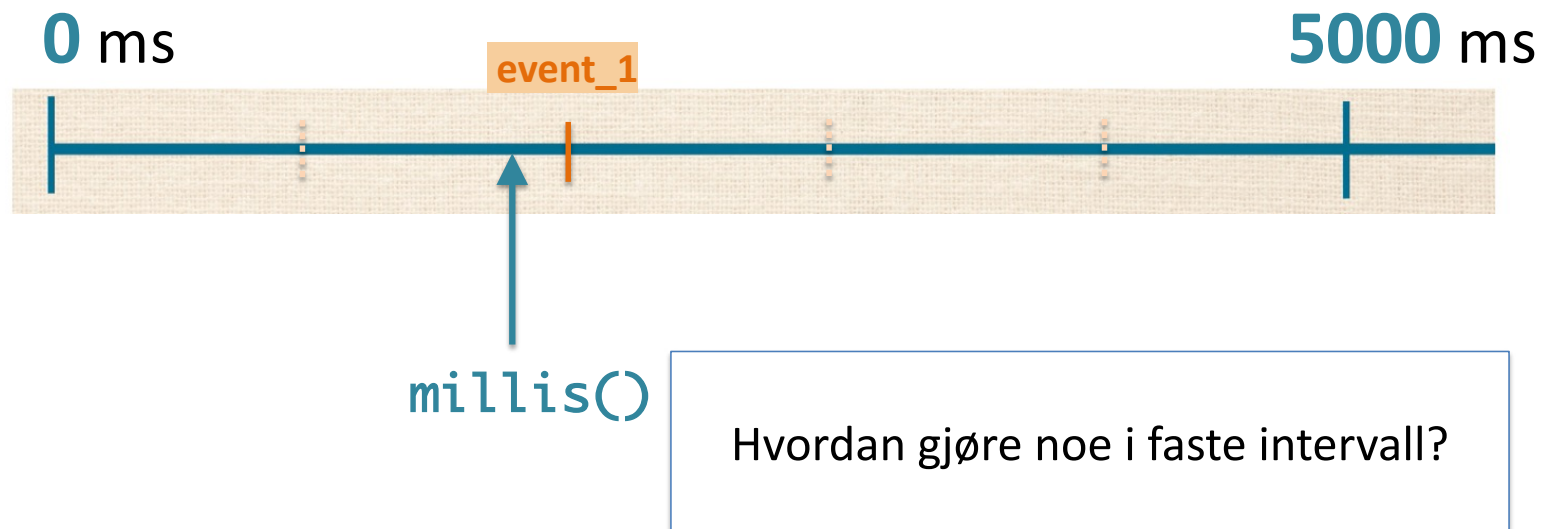
# Repeterende Timede Hendelser (RTH)



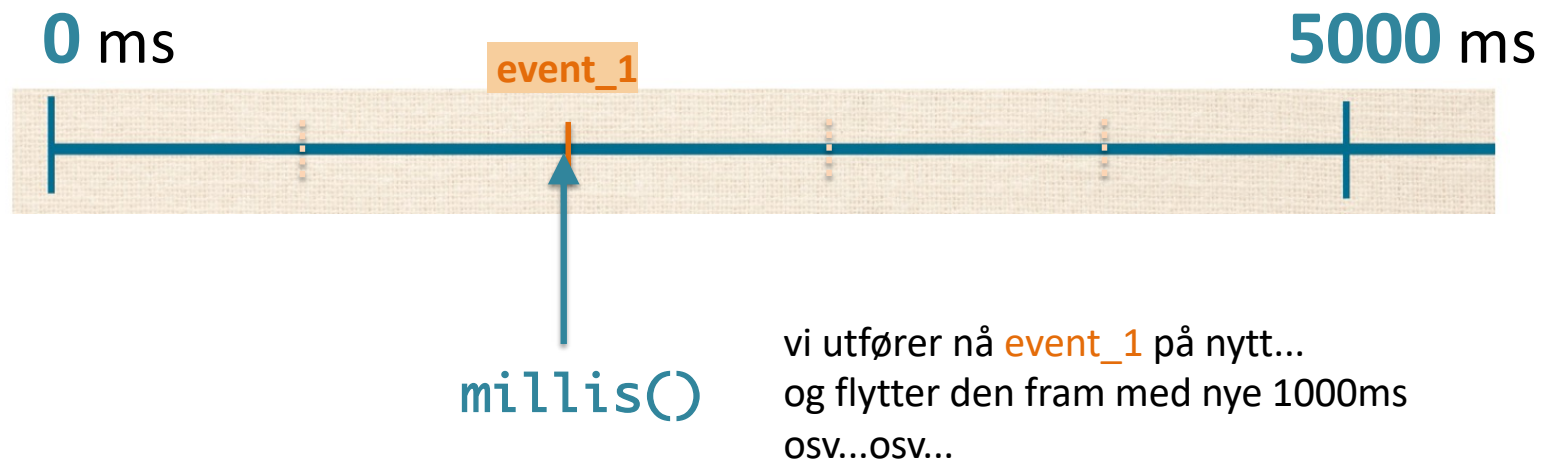
# Repeterende Timede Hendelser (RTH)



# Repeterende Timede Hendelser (RTH)



# Repeterende Timede Hendelser (RTH)





# Repeterende Timede Hendelser (RTH)

**millis()**

*Teller antall  
millisekunder*

```
1 const unsigned long eventInterval = 1000;  
2  
3 void setup(){  
4  
5 }  
6  
7 void loop(){  
8  
9 }
```

# Repeterende Timede Hendelser (RTH)

**millis()**

*Teller antall  
millisekunder*

```
1 const unsigned long eventInterval = 1000;
2 unsigned long previousTime = 0;
3
4 void setup(){
5
6 }
7
8 void loop(){
9
10 }
```

# Repeterende Timede Hendelser (RTH)

**millis()**

*Teller antall  
millisekunder*

```
1 const unsigned long eventInterval = 1000;
2 unsigned long previousTime = 0;
3
4 void setup() {
5     Serial.begin(9600);
6 }
7
8 void loop() {
9
10     /* Updates frequently */
11     unsigned long currentTime = millis();
12
13 }
```

# RTH

## millis()

*Teller antall  
millisekunder*

```
1 const unsigned long eventInterval = 1000;
2 unsigned long previousTime = 0;
3
4 void setup() {
5   Serial.begin(9600);
6 }
7
8 void loop() {
9
10  /* Updates frequently */
11  unsigned long currentTime = millis();
12
13  /* This is the event*/
14  if ( currentTime - previousTime >= eventInterval) {
15    /* Event code*/
16    Serial.println("testing ...");
17
18    /* Update the timing for the next time around */
19    previousTime = currentTime;
20  }
21 }
```

# RTH

**millis()**

*Teller antall  
millisekunder*

```
11 unsigned long currentTime = millis();
12
13 /* This is the event*/
14 if ( currentTime - previousTime >= eventInterval) {
15     /* Event code*/
16     Serial.println("testing ...");
17
18     /* Update the timing for the next time around */
19     previousTime = currentTime;
```