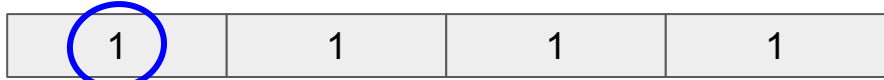


Duration vs. Timestamps

Cumulatieve afwijking



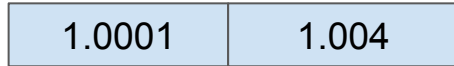
`time.sleep(dur)`

1.0001

Cumulatieve afwijking



`time.sleep(dur)`



Cumulatieve afwijking

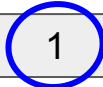


```
time.sleep(dur)
```



Cumulatieve afwijking

1	1	1	1
---	---	---	---



```
time.sleep(dur)
```

1.0001	1.004	1.002	1.03
--------	-------	-------	------

Cumulatieve afwijking

1	1	1	1
---	---	---	---

```
time.sleep(dur)
```

1.0001	1.004	1.002	1.03
--------	-------	-------	------

afwijking

Cumulatieve afwijking

1	1	1	1
---	---	---	---

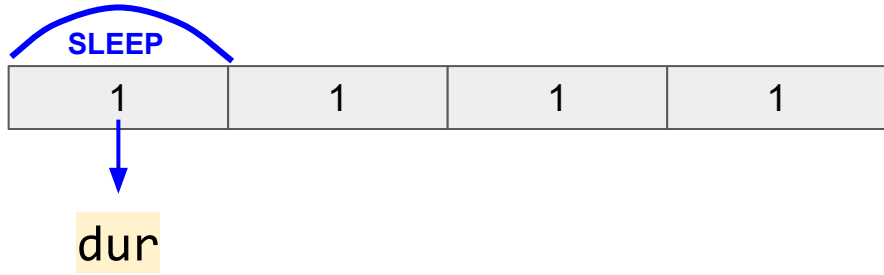
```
time.sleep(dur)
```

1.0001	1.004	1.002	1.03
--------	-------	-------	------

afwijking

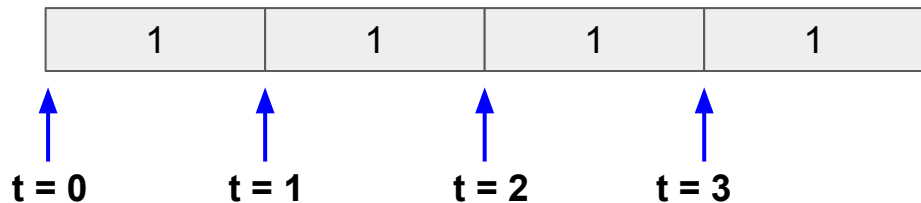
Hoe te voorkomen?
Iemand een idee?

Duration



```
for dur in sequence:  
    sample.play()  
    time.sleep(dur)
```


Duration → Timestamps



```
// TODO - play sample at  
// t == 0 || t == 1 ||  
// t == 2 || t == 3
```

Timestamps



```
timestamp_seq = [ 0, 1, 2, 3 ]
```

initialize timestamp sequence

Timestamps



```
timestamp_seq = [ 0, 1, 2, 3 ]
```

```
ts = timestamp_seq[0]
```

retrieve first timestamp

Timestamps



```
timestamp_seq = [ 0, 1, 2, 3]
```

```
if t >= ts: TRUE  
    sample.play()  
    ts = timestamp_seq[nextIndex]  
  
time.sleep(0.001)
```

Timestamps



```
timestamp_seq = [ 0, 1, 2, 3]
```

```
if t >= ts:
```

```
    sample.play()
```

```
    ts = timestamp_seq[nextIndex]
```

```
time.sleep(0.001)
```

play sample

Timestamps



t = 0.000

```
timestamp_seq = [ 0, 1, 2, 3 ]
```

```
if t >= ts:  
    sample.play()  
    ts = timestamp_seq[nextIndex]
```

retrieve the next timestamp

```
time.sleep(0.001)
```

Timestamps



```
timestamp_seq = [ 0, 1, 2, 3]
```

```
if t >= ts:  
    sample.play()  
    ts = timestamp_seq[nextIndex]
```

```
time.sleep(0.001)
```

minor fraction 'forward in time'

Timestamps



```
timestamp_seq = [ 0, 1, 2, 3]
```

```
if t >= ts:  
    sample.play()  
    ts = timestamp_seq[nextIndex]
```

```
time.sleep(0.001)
```


Timestamps



t = 0.001

```
timestamp_seq = [ 0, 1, 2, 3 ]
```

```
if t >= ts:  
    sample.play()  
    ts = timestamp_seq[nextIndex]
```

FALSE

```
time.sleep(0.001)
```

Timestamps



$t = 0.001$

```
timestamp_seq = [ 0, 1, 2, 3 ]
```

```
if t >= ts:  
    sample.play()  
    ts = timestamp_seq[nextIndex]
```

```
time.sleep(0.001)
```

minor fraction 'forward in time'

Timestamps



t = 0.002

```
timestamp_seq = [ 0, 1, 2, 3]
```

```
if t >= ts:  
    sample.play()  
    ts = timestamp_seq[nextIndex]
```

```
time.sleep(0.001)
```

Timestamps



```
timestamp_seq = [ 0, 1, 2, 3 ]
```

```
if t >= ts:  
    sample.play()  
    ts = timestamp_seq[nextIndex]  
  
time.sleep(0.001)
```

FALSE

Timestamps



$t = 0.002$

```
timestamp_seq = [ 0, 1, 2, 3 ]
```

```
if t >= ts:  
    sample.play()  
    ts = timestamp_seq[nextIndex]
```

```
time.sleep(0.001)
```

minor fraction 'forward in time'

Timestamps



t = 0.003

```
timestamp_seq = [ 0, 1, 2, 3]
```

```
if t >= ts:  
    sample.play()  
    ts = timestamp_seq[nextIndex]
```

```
time.sleep(0.001)
```

Timestamps



```
timestamp_seq = [ 0, 1, 2, 3 ]
```

```
if t >= ts:  
    sample.play()  
    ts = timestamp_seq[nextIndex]  
  
time.sleep(0.001)
```

FALSE

Timestamps



```
timestamp_seq = [ 0, 1, 2, 3 ]
```

```
if t >= ts:  
    sample.play()  
    ts = timestamp_seq[nextIndex]
```

```
time.sleep(0.001)
```

minor fraction 'forward in time'

Timestamps



...

```
timestamp_seq = [ 0, 1, 2, 3]
```

```
if t >= ts:  
    sample.play()  
    ts = timestamp_seq[nextIndex]
```

```
time.sleep(0.001)
```

Timestamps



$t = 1.000$

```
timestamp_seq = [ 0, 1, 2, 3]
```

```
if t >= ts:  
    sample.play()  
    ts = timestamp_seq[nextIndex]
```

```
time.sleep(0.001)
```

Timestamps



```
timestamp_seq = [ 0, 1, 2, 3]
```

```
if t >= ts: TRUE  
    sample.play()  
    ts = timestamp_seq[nextIndex]  
  
time.sleep(0.001)
```

Timestamps



```
timestamp_seq = [ 0, 1, 2, 3 ]
```

```
if t >= ts:
```

```
    sample.play()
```

```
    ts = timestamp_seq[nextIndex]
```

```
time.sleep(0.001)
```

play sample

Timestamps



```
timestamp_seq = [ 0, 1, 2, 3 ]
```

```
if t >= ts:  
    sample.play()  
    ts = timestamp_seq[nextIndex]
```

```
time.sleep(0.001)
```

retrieve the next timestamp

Timestamps



```
timestamp_seq = [ 0, 1, 2, 3 ]
```

```
if t >= ts:  
    sample.play()  
    ts = timestamp_seq[nextIndex]
```

```
time.sleep(0.001)
```

minor fraction 'forward in time'

Timestamps



etcetera

```
timestamp_seq = [ 0, 1, 2, 3 ]
```

```
if t >= ts:  
    sample.play()  
    ts = timestamp_seq[nextIndex]
```

```
time.sleep(0.001)
```

Timestamps

```
timestamp_seq = [ 0, 1, 2, 3 ]  
ts = timestamp_seq.pop(0)
```

```
...  
// retrieve current_ts  
...  
if current_ts >= ts:  
    sample.play()  
    ts = timestamp_seq.pop(0)  
time.sleep(0.001)  
...  
// repeat the above
```


Multiple samples

```
# first item in sublist is the timestamp, second is  
the sample index
```

```
event_seq = [[0, 0], [0.5, 1], [1.5, 0], [3.0, 1]]
```

```
event = event_seq.pop(0)
```

```
event[0] → timestamp
```

```
event[1] → bevat the sample index
```

```
# dictionary = duidelijk
```