

# **Forelesning IN1900 – 5 okt 2023**

**Ole Christian Lingjærde  
Institutt for Informatikk, Universitetet i Oslo**

**Uke: 2 Oktober - 8 Oktober, 2023**

# Denne ukens agenda

Dictionaries (mandag)

Tekststrenger (i dag)

Oppvarming til midtveiseksamen (i dag)

## Tekstbehandling - kjapp repetisjon

```
s = "min fil.txt"
```

```
# Splitte i enkeltord:
```

```
s.split()    # ['min', 'fil.txt']
```

# Tekstbehandling - kjapp repetisjon

```
s = "min fil.txt"
```

```
# Splitte i enkeltord:
```

```
s.split()    # ['min', 'fil.txt']
```

```
# Splitte i enkeltord med selvvalgt skilletegn:
```

```
s.split(".")    # ['min fil', 'txt']
```

```
s.split(" fil")    # ['min', '.txt']
```

## Tekstbehandling - kjapp repetisjon

```
s = "min fil.txt"
```

```
# Splitte i enkeltord:
```

```
s.split()    # ['min', 'fil.txt']
```

```
# Splitte i enkeltord med selvvalgt skilletegn:
```

```
s.split(".")    # ['min fil', 'txt']
```

```
s.split(" fil")    # ['min', '.txt']
```

```
# Finne plassering av substreng:
```

```
s.index(".")    # 7
```

## Tekstbehandling - kjapp repetisjon

```
s = "min fil.txt"
```

```
# Splitte i enkeltord:
```

```
s.split()    # ['min', 'fil.txt']
```

```
# Splitte i enkeltord med selvvalgt skilletegn:
```

```
s.split(".")    # ['min fil', 'txt']
```

```
s.split(" fil") # ['min', '.txt']
```

```
# Finne plassering av substreng:
```

```
s.index(".")    # 7
```

```
# Sjekke om substreng er tilstede:
```

```
"fil" in s      # True
```

```
"FIL" in s      # False
```

# Tekstbehandling - kjapp repetisjon

```
s = "min fil.txt"
```

```
# Splitte i enkeltord:
```

```
s.split()    # ['min', 'fil.txt']
```

```
# Splitte i enkeltord med selvvalgt skilletegn:
```

```
s.split(".")    # ['min fil', 'txt']
```

```
s.split(" fil") # ['min', '.txt']
```

```
# Finne plassering av substreng:
```

```
s.index(".")    # 7
```

```
# Sjekke om substreng er tilstede:
```

```
"fil" in s      # True
```

```
"FIL" in s      # False
```

```
# Plukke ut ett tegn:
```

```
s[0]           # 'm'
```

```
s[1]           # 'i'
```

```
s[2]           # 'n'
```

```
s[3]           # ' '
```

# Plukke ut substreng

```
s = "Dette er en tekststreng"
```

```
# Alt unntatt første tegn
```

```
s[1:] # "ette er en tekststreng"
```



## Plukke ut substreng

```
s = "Dette er en tekststreng"
```

```
# Alt unntatt første tegn
```

```
s[1:] # "ette er en tekststreng"
```

```
# Alt unntatt første og siste tegn
```

```
s[1:-1] # "ette er en tekststren"
```

# Plukke ut substreng

```
s = "Dette er en tekststreng"
```

```
# Alt unntatt første tegn
```

```
s[1:] # "ette er en tekststreng"
```

```
# Alt unntatt første og siste tegn
```

```
s[1:-1] # "ette er en tekststren"
```

```
# Tegnene med indeks 2,3,4
```

```
s[2:5] # "tte"
```

# Plukke ut substreng

```
s = "Dette er en tekststreng"

# Alt unntatt første tegn
s[1:] # "ette er en tekststreng"

# Alt unntatt første og siste tegn
s[1:-1] # "ette er en tekststren"

# Tegnene med indeks 2,3,4
s[2:5] # "tte"

# Alt fra og med en substreng
s[s.index("tekst"):] # "tekststreng"
```

# Plukke ut substreng

```
s = "Dette er en tekststreng"

# Alt unntatt første tegn
s[1:] # "ette er en tekststreng"

# Alt unntatt første og siste tegn
s[1:-1] # "ette er en tekststren"

# Tegnene med indeks 2,3,4
s[2:5] # "tte"

# Alt fra og med en substreng
s[s.index("tekst"):] # "tekststreng"

# Fjern blanke foran og bak
s = "      A B C      "
s.strip() # "A B C"
s.lstrip() # "A B C      "
s.rstrip() # "      A B C"
```

# Slå sammen tekststrenger

```
a = ["I", "am", "happy"]
```

```
# Slå sammen listeelementer
```

```
s = "".join(a)    # "Iamhappy"
```

## Slå sammen tekststrenger

```
a = ["I", "am", "happy"]
```

```
# Slå sammen listeelementer
```

```
s = "".join(a)    # "Iamhappy"
```

```
# Slå sammen listeelementer med blanke i mellom
```

```
s = " ".join(a)  # "I am happy"
```

## Slå sammen tekststrenger

```
a = ["I", "am", "happy"]
```

```
# Slå sammen listeelementer
```

```
s = "".join(a) # "Iamhappy"
```

```
# Slå sammen listeelementer med blanke i mellom
```

```
s = " ".join(a) # "I am happy"
```

```
# Slå sammen listeelementer med "--" i mellom
```

```
s = "--".join(a) # "I--am--happy"
```

## Bytt ut substreng

```
s = "Dette er en tekststreng"
```

```
# Erstatt alle blanke med "X"
```

```
t = s.replace(" ", "X")
```

```
# 'DetteXerXenXtekststreng'
```



## Bytt ut substreng

```
s = "Dette er en tekststreng"
```

```
# Erstatt alle blanke med "X"
```

```
t = s.replace(" ", "X")
```

```
# 'DetteXerXenXtekststreng'
```

```
# Erstatt en substreng med en annen
```

```
t = s.replace("Dette", "Her")
```

```
# 'Her er en tekststreng'
```

## Bytt ut substreng

```
s = "Dette er en tekststreng"
```

```
# Erstatt alle blanke med "X"
```

```
t = s.replace(" ", "X")
```

```
  # 'DetteXerXenXtekststreng'
```

```
# Erstatt en substreng med en annen
```

```
t = s.replace("Dette", "Her")
```

```
  # 'Her er en tekststreng'
```

```
# Erstatt alt foran "tekst" med noe annet
```

```
t = s.replace(s[:s.index("tekst")], "Ny ")
```

```
  # 'Ny tekststreng'
```

# Bytt ut substreng

```
s = "Dette er en tekststreng"
```

```
# Erstatt alle blanke med "X"
```

```
t = s.replace(" ", "X")  
# 'DetteXerXenXtekststreng'
```

```
# Erstatt en substreng med en annen
```

```
t = s.replace("Dette", "Her")  
# 'Her er en tekststreng'
```

```
# Erstatt alt foran "tekst" med noe annet
```

```
t = s.replace(s[:s.index("tekst")], "Ny ")  
# 'Ny tekststreng'
```

```
# Erstatt alt fra og med "tekst" med noe annet
```

```
t = s.replace(s[s.index("tekst"):], "setning")  
# 'Dette er en setning'
```

# Ny linje i tekststrenger

Her er det forskjeller mellom operativsystemer!

***# Unix/Linux/Mac:***

```
s = "\n".join(["Line A", "Line B", "Line C"])
s.split("\n")
```

***# Windows:***

```
s = "\r\n".join(["Line A", "Line B", "Line C"])
s.split("\r\n")
```

***# Alle operativsystemer:***

```
s.splitlines()
```

# Noen flere tekstfunksjoner

*# Test om en string bare består av sifre*

```
s = "314"
```

```
s.isdigit()    # True
```

```
s = " 314"
```

```
s.isdigit()    # False
```

```
s = "3.14"
```

```
s.isdigit()    # False
```

## Noen flere tekstfunksjoner

*# Test om en string bare består av sifre*

```
s = "314"
```

```
s.isdigit()    # True
```

```
s = " 314"
```

```
s.isdigit()    # False
```

```
s = "3.14"
```

```
s.isdigit()    # False
```

*# Endre alt til små eller til store bokstaver*

```
s = "ABC def"
```

```
s.lower()     # "abc def"
```

```
s.upper()     # "ABC DEF"
```

# Noen flere tekstfunksjoner

## *# Test om en string bare består av sifre*

```
s = "314"  
s.isdigit()    # True  
s = " 314"  
s.isdigit()    # False  
s = "3.14"  
s.isdigit()    # False
```

## *# Endre alt til små eller til store bokstaver*

```
s = "ABC def"  
s.lower()     # "abc def"  
s.upper()     # "ABC DEF"
```

## *# Test om en string starter/slutter med en gitt string*

```
s = "Dette er en string"  
s.startswith("Dette er")    # True  
s.endswith("Dette er")      # False
```

# Noen flere tekstfunksjoner

*# Teste om en string bare består av bokstaver:*

```
s = "Vinter, vår og høst"  
s.isalpha()          # False  
s[0:5].isalpha()    # True
```

*# Fjerne alle spesialtegn:*

```
s0 = "Vinter, vår og høst"  
s1 = ""  
for e in s0:  
    if e.isalpha():  
        s1 = s1 + e  
# Nå er s1: "Vintervårøghøst"
```



## Eksempel: Ekstrahere tall i en tekstfil

Anta at vi ønsker å lese en fil med følgende format:

```
(1.3, 0)    (-1, 2)    (3, -1.5)
(0, 1)     (1, 0)    (1, 1)
(0, -0.01) (10.5, -1) (2.5, -2.5)
```

Algoritme:

- 1 Les én linje av gangen
- 2 For hver linje: splitt opp i ord
- 3 For hvert ord: fjern parenteser og splitt på komma

# Implementasjon

```
pairs = []

with open('pairs.dat') as infile:
    for line in infile:
        words = line.split()
        for w in words:
            w = w[1:-1]    # Fjern parentesene rundt
            numbers = w.split(",")
            pair = (float(numbers[0]), float(numbers[1]))
            pairs.append(pair)
```

pairs:

```
[(1.3, 0.0),  
 (-1.0, 2.0),  
 (3.0, -1.5),  
 (0.0, 1.0),  
 (1.0, 0.0),  
 (1.0, 1.0),  
 (0.0, -0.01),  
 (10.5, -1.0),  
 (2.5, -2.5)]
```

## Eksempel: Finne oftest forekommende ord

Vi ønsker å lage et program som finner de oftest forekommende ordene i en tekstfil.

Vi ønsker å lage et program som finner de oftest forekommende ordene i en tekstfil.

### **Innledende strategi:**

- Lag en dictionary words som lagrer ordene som nøkler og antall forekomster som verdier
- Les filen linjevis med `readline()`
- Splitt hver linje i enkeltord med `split()`
- Legg enkeltordene inn i `words`

## Eksempel: Finne vanligst forekommende ord

**Problem 1: Hvordan unngå at spesialtegn og store/små bokstaver skaper problemer?**

## Eksempel: Finne vanligst forekommende ord

### Problem 1: Hvordan unngå at spesialtegn og store/små bokstaver skaper problemer?

Anta vi har tekstfilen:

```
Sola skinner. Det er veldig varmt i sola i dag. Hvis  
det ikke hadde vært for sola, ville det ha vært kaldt.
```

Her vil `split()` fange opp blant annet ordene

```
Sola sola sola,
```

og vi ønsker å registrere dette som samme ord.

## Eksempel: Finne vanligst forekommende ord

### Problem 1: Hvordan unngå at spesialtegn og store/små bokstaver skaper problemer?

Anta vi har tekstfilen:

```
Sola skinner. Det er veldig varmt i sola i dag. Hvis  
det ikke hadde vært for sola, ville det ha vært kaldt.
```

Her vil `split()` fange opp blant annet ordene

```
Sola sola sola,
```

og vi ønsker å registrere dette som samme ord.

#### Løsning:

- 1) Trekk ut bokstavene i hvert ord med `isalpha()`
- 2) Konverter til små bokstaver med `lower()`



Eksempel: Finne vanligst forekommende ord

**Problem 2: Hvordan telle antall forekomster av hvert ord?**

## Problem 2: Hvordan telle antall forekomster av hvert ord?

### Løsning:

- Lag en tom dictionary  $counts = \{ \}$
- For alle ordene  $w$  i filen:
  - Hvis ordet ikke er sett før:  
 $counts[w] = 1$
  - Hvis ordet er sett før:  
 $counts[w] = counts[w] + 1$

## Eksempel: Finne vanligst forekommende ord

```
filename = input("File name? ")
file = open(filename)

# Les filen linjevis, splitt linjer i ord og legg inn i dictionary
counts = {}
for line in file:
    words = line.split()
    for w in words:
        w = "".join([e for e in w if e.isalpha()])
        w = w.lower()
        if w in counts:
            counts[w] = counts[w]+1
        else:
            counts[w] = 1

# Put all words in list and sort on counts
words = list(counts.keys())
words.sort(key=lambda x:counts[x], reverse=True)

# Print 30 most frequent words with counts
for i in range(0,min(len(words),30)):
    print(f"{i+1:3d} {words[i]:10s} {counts[words[i]]}")
```

Onsdag 11. oktober kl 15-19.

Sted:

- Silurveien 2 Sal 3B
- Silurveien 2 Sal 3C
- Silurveien 2 Enerom 4
- Silurveien 2 Enerom 5
- Silurveien 2 Sal 3D

Sjekk i Studentweb hvor du skal sitte, og møt opp i god tid!

Noen reisealternativer:

- T-banen: linje 3 til Åsjordet, deretter ca 600 meter å spasere
- Buss: linje 23 til Blokkajordet, deretter ca 300 meter å spasere

Oppgavesettet består av flervalgsoppgaver.

Oppgaver kan være av formen

"Velg ett alternativ"

Riktig svar gir 1 poeng

Feil svar gir 0 poeng

Manglende svar gir 0 poeng

"Hvilke alternativer/verdier er riktige"

Hvert riktig svar gir 0.5 poeng

Hvert feil svar gir 0 poeng

Hvert manglende svar gir 0 poeng

## i Forside

UNIVERSITY OF OSLO

Faculty of mathematics and natural sciences

Mid-term exam IN1900, MAT-IN1105, IN-KJM1900, IN-GEO1900

Date: October 12, 2022

Attachments: None

Permitted aids: None

- The exam set has 21 multiple choice questions. The maximum score is 25.
- All code in the question text is written in Python 3.
- Most of the questions have one correct answer. A correctly marked answer gives 1 point, wrong or missing answer gives 0 points. There are no negative points for wrong answers, so one should always mark an answer.
- Questions 16 and 17 have 4 answers each. It is given 0.5 points for each correct answer and 0 points for wrong or unmarked answers, maximum 2 points total for each of the two questions.
- Questions 18 and 21 have two correct answers. It is given 1 point for each correctly marked answer, -1 for incorrectly marked answers, and 0 for missing answers. The maximum score on the question is 2 and the minimum score is 0. One should always mark at least one answer.

## 1 Hva skrives ut?

What is printed in the terminal when the following code is run?

```
a = 3
```

```
b = a + 2
```

```
a = 100
```

```
print(a,b)
```

Select one alternative:

- 3 5
- 100 102
- 100 5
- An error message

## 1 Hva skrives ut?

What is printed in the terminal when the following code is run?

```
a = 3
```

```
b = a + 2
```

```
a = 100
```

```
print(a,b)
```

Select one alternative:

3 5

100 102

100 5

An error message



## 2 Hva skrives ut?

What is printed when this code is run?

```
x = [4,5,6]
```

```
y = [1,2,3]
```

```
print(x + y)
```

Select one alternative:

- An error message
- [4,5,6,1,2,3]
- [1,2,3,4,5,6]
- [5,7,9]

---

Maximum marks: 1

## 2 Hva skrives ut?

What is printed when this code is run?

```
x = [4,5,6]
```

```
y = [1,2,3]
```

```
print(x + y)
```

Select one alternative:

An error message

[4,5,6,1,2,3]

[1,2,3,4,5,6]

[5,7,9]

### 3 Hva skrives ut?

What is printed when this code is run?

```
x = 0
a = 1
b = 4
if a - b < 0:
    if b != 0:
        x = x + 1
    if b > 0:
        x = x + 2
    elif a > 5:
        x = x + 3
    else:
        x = x + 4
else:
    x = x + 5
```

print(x)

Select one alternative:

- 2
- 5
- 10
- 3
- 1

## 3 Hva skrives ut?

What is printed when this code is run?

```
x = 0
a = 1
b = 4
if a - b < 0:
    if b != 0:
        x = x + 1
    if b > 0:
        x = x + 2
    elif a > 5:
        x = x + 3
    else:
        x = x + 4
else:
    x = x + 5
```

print(x)

Select one alternative:

- 2
- 5
- 10
- 3
- 1

$$x = 0, a = 1, b = 4$$

$$a - b < 0$$

$$b \neq 0$$

$$x = 1$$

$$b > 0$$

$$x = 3$$

#### 4 Hva skrives ut?

What is printed when this code is run?

```
import math
```

```
r = 1
```

```
A = math.pi*r**2
```

```
print(type(r),type(A))
```

Select one alternative:

- An error message
- 1.0 3.14
- <class 'int'> <class 'float'>
- <class 'float'> <class 'float'>

---

Maximum marks: 1

#### 4 Hva skrives ut?

What is printed when this code is run?

```
import math
```

```
r = 1
```

```
A = math.pi*r**2
```

```
print(type(r),type(A))
```

Select one alternative:

- An error message
- 1.0 3.14
- <class 'int'> <class 'float'>
- <class 'float'> <class 'float'>

---

Maximum marks: 1

## 5 Hva skrives ut?

What is printed when this code is run?

```
a = []  
x = []  
for n in range(100):  
    x.append(n)  
    a.append(x)  
print(len(a), len(a[-1]))
```

Select one alternative:

- 100 100
- 100 1
- An error message
- 101 1
- 101 101

## 5 Hva skrives ut?

What is printed when this code is run?

```
a = []
x = []
for n in range(100):
    x.append(n)
    a.append(x)
print(len(a), len(a[-1]))
```

Select one alternative:

- 100 100
- 100 1
- An error message
- 101 1
- 101 101



## 6 Hva skrives ut?

What is printed when this code is run?

```
hello = "Hello students! Welcome to the midterm exam!"  
print(hello.split('e')[-1])
```

Select one alternative:

- An error message
- exam!
- xam!
- Welcome to the midterm exam!

---

Maximum marks: 1

## 6 Hva skrives ut?

What is printed when this code is run?

```
hello = "Hello students! Welcome to the midterm exam!"  
print(hello.split('e')[-1])
```

Select one alternative:

- An error message
- exam!
- xam!
- Welcome to the midterm exam!

---

Maximum marks: 1

## 7 Hva skrives ut?

What is printed when this code is run?

```
s = 0
for i in range(2,8,2):
    s += i
print(s)
```

Select one alternative:

- 0
- 22
- 20
- 12

---

Maximum marks: 1

## 7 Hva skrives ut?

What is printed when this code is run?

```
s = 0
for i in range(2,8,2):
    s += i
print(s)
```

Select one alternative:

- 0
- 22
- 20
- 12

---

Maximum marks: 1

## 8 Hva skrives ut?

What is printed when this code is run?

```
def f(x,y):  
    return 3*x**2 - 4*y
```

```
y = 3  
x = 2  
print(f(1,2))
```

Select one alternative:

- 5
- 10
- An error message
- 0

---

Maximum marks: 1

## 8 Hva skrives ut?

What is printed when this code is run?

```
def f(x,y):  
    return 3*x**2 - 4*y
```

```
y = 3  
x = 2  
print(f(1,2))
```

Select one alternative:

- 5
- 10
- An error message
- 0

---

Maximum marks: 1

## 9 Hva skrives ut?

What is printed when this code is run?  
`from math import sqrt`

```
def norm(v):  
    v_squared = [e**2 for e in v]  
    return sqrt(sum(v_squared))  
  
def test_norm():  
    v = (4,3)  
    expected = 5  
    computed = norm(v)  
    tol = 1e-6  
    msg = f'expected {expected}, got {computed}'  
    assert abs(expected -computed) < tol, msg
```

`test_norm()`

The function `sum`, which is used inside `norm`, is a built-in function in Python. It takes a list, array, or similar object as argument, and returns the sum of its elements.

Select one alternative:

- False
- AssertionError: expected 5, got 0
- Nothing is printed
- True

## 9 Hva skrives ut?

What is printed when this code is run?  
`from math import sqrt`

```
def norm(v):  
    v_squared = [e**2 for e in v]  
    return sqrt(sum(v_squared))  
  
def test_norm():  
    v = (4,3)  
    expected = 5  
    computed = norm(v)  
    tol = 1e-6  
    msg = f'expected {expected}, got {computed}'  
    assert abs(expected -computed) < tol, msg
```

`test_norm()`

The function `sum`, which is used inside `norm`, is a built-in function in Python. It takes a list, array, or similar object as argument, and returns the sum of its elements.

Select one alternative:

- False
- AssertionError: expected 5, got 0
- Nothing is printed
- True



**10 Hva skrives ut?**

What is printed when this code is run?

```
F = 0
```

```
C1 = []; C2 = []
```

```
while F <= 100:
```

```
    C1.append(5/9 * (F-32))
```

```
    F += 10
```

```
for F in range(0,100,10):
```

```
    C2.append(5/9 * (F-32))
```

```
print(F, len(C1) == len(C2))
```

Select one alternative:

- 90 False
- 100 True
- 90 True
- 100 False

## 10 Hva skrives ut?

What is printed when this code is run?

```
F = 0
```

```
C1 = []; C2 = []
```

```
while F <= 100:
```

```
    C1.append(5/9 * (F-32))
```

```
    F += 10
```

```
for F in range(0,100,10):
```

```
    C2.append(5/9 * (F-32))
```

```
print(F, len(C1) == len(C2))
```

Select one alternative:

90 False

100 True

90 True

100 False

## 11 I hvilken linje feiler koden?

In what line will this code stop and print an error message?

```
n = 100
x = [0]*(n+1)
dx = 1.0/n
for i in range(n+1):
    x[i+1] = i * dx
```

Select one alternative:

- x[i+1] = i \* dx
- for i in range(n+1):
- dx = 1.0/n
- x = [0]\*(n+1)

---

Maximum marks: 1

## 11 I hvilken linje feiler koden?

In what line will this code stop and print an error message?

```
n = 100
x = [0]*(n+1)
dx = 1.0/n
for i in range(n+1):
    x[i+1] = i * dx
```

Select one alternative:

- `x[i+1] = i * dx`
- `for i in range(n+1):`
- `dx = 1.0/n`
- `x = [0]*(n+1)`

---

Maximum marks: 1

**12 I hvilken linje feiler koden?**

In what line will this code stop and print an error message?

```
def g(x):  
    return x**3 + x**2
```

```
def f(x,y):  
    return x + 2*y
```

```
x = 1  
print(f(x,g(x)))  
print(g(y))
```

Select one alternative:

- print(f(x,g(x)))
- return x + 2\*y
- return x\*\*3 + x\*\*2
- print(g(y))

## 12 I hvilken linje feiler koden?

In what line will this code stop and print an error message?

```
def g(x):  
    return x**3 + x**2
```

```
def f(x,y):  
    return x + 2*y
```

```
x = 1  
print(f(x,g(x)))  
print(g(y))
```

Select one alternative:

- print(f(x,g(x)))
- return x + 2\*y
- return x\*\*3 + x\*\*2
- print(g(y))