

Samlinger, lister

Læringsmål

Læringsmål

- Etter modulen:

Læringsmål

- Etter modulen:
 - Forstår du intuisjonen og formålet bak bruk av samlinger for å holde på mange verdier

Læringsmål

- Etter modulen:
 - Forstår du intuisjonen og formålet bak bruk av samlinger for å holde på mange verdier
 - Ser du hvordan du kan ta i bruk samlinger hvor det kreves for å løse et programmeringsproblem

Læringsmål

- Etter modulen:
 - Forstår du intuisjonen og formålet bak bruk av samlinger for å holde på mange verdier
 - Ser du hvordan du kan ta i bruk samlinger hvor det kreves for å løse et programmeringsproblem
 - Kjenner du til tjenestene en liste tilbyr og kan bruke disse i egne programmer

Forkunnskaper

Forkunnskaper

- Modulen bygger direkte på:

Forkunnskaper

- Modulen bygger direkte på:
 - Variabler

Forkunnskaper

- Modulen bygger direkte på:
 - Variabler
 - Evaluering av uttrykk og utførelse av kodelinjer

Forkunnskaper

- Modulen bygger direkte på:
 - Variabler
 - Evaluering av uttrykk og utførelse av kodelinjer
- Nyttig for å forstå alle aspekter og eksempler:

Forkunnskaper

- Modulen bygger direkte på:
 - Variabler
 - Evaluering av uttrykk og utførelse av kodelinjer
- Nyttig for å forstå alle aspekter og eksempler:
 - Beslutninger

Sjonglere med flere
verdier

Sjonglere med flere verdier

- {hoyde1.py}

Finne verdien vi trenger direkte

```
hoydeAar0 = 50  
hoydeAar1 = 76  
hoydeAar2 = 87  
hoydeAar3 = 96
```

```
alder = int(input("Hvilken alder vil du vite hoyden  
for (0,1,2 eller 3 aar)? "))
```

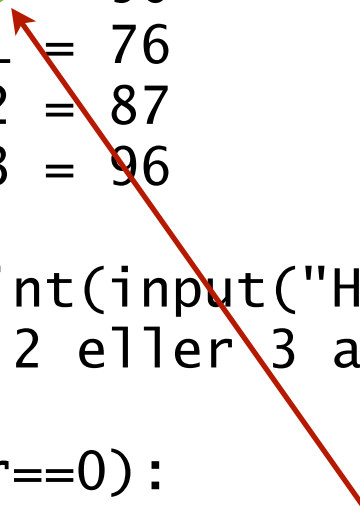
```
if (alder==0):  
    print(hoydeAar0)  
elif (alder==1):  
    print(hoydeAar1)  
elif (alder==2):  
    print(hoydeAar2)  
elif (alder==3):  
    print(hoydeAar3)
```

Finne verdien vi trenger direkte

```
hoydeAar0 = 50  
hoydeAar1 = 76  
hoydeAar2 = 87  
hoydeAar3 = 96
```

```
alder = int(input("Hvilken alder vil du vite hoyden  
for (0,1,2 eller 3 aar)? "))
```

```
if (alder==0):  
    print(hoydeAaralder)  
elif (alder==1):  
    print(hoydeAar1)  
elif (alder==2):  
    print(hoydeAar2)  
elif (alder==3):  
    print(hoydeAar3)
```



Vi kan slå opp verdien
vi trenger direkte!

Vi kan slå opp verdien vi trenger direkte!

- Det vi ønsket:

Vi kan slå opp verdien vi trenger direkte!

- Det vi ønsket:
 - høydeAar**alder**

Vi kan slå opp verdien vi trenger direkte!

- Det vi ønsket:
 - hoydeAar**alder**
- Syntaks i Python:

Vi kan slå opp verdien vi trenger direkte!

- Det vi ønsket:
 - `hoydeAar`**alder**
- Syntaks i Python:
 - `hoydeAar[alder]`

Vi kan slå opp verdien vi trenger direkte!

- Det vi ønsket:
 - hoydeAar**alder**
- Syntaks i Python:
 - hoydeAar[alder]
- Og før dette må vi definere hoydeAar som en liste:

Vi kan slå opp verdien vi trenger direkte!

- Det vi ønsket:
 - hoydeAar**alder**
- Syntaks i Python:
 - hoydeAar[alder]
- Og før dette må vi definere hoydeAar som en liste:
 - ```
hoydeAar = [50, 76, 87, 96]
```

Håndtere høydene i en  
liste



# Håndtere høydene i en liste

- {hoyde2.py}

# Liste

- Definere en liste:

-

# Liste

- Definere en liste:
  - hoydeAar = [50, 76, 87, 96]

|   |    |
|---|----|
| 0 | 50 |
| 1 | 76 |
| 2 | 87 |
| 3 | 96 |

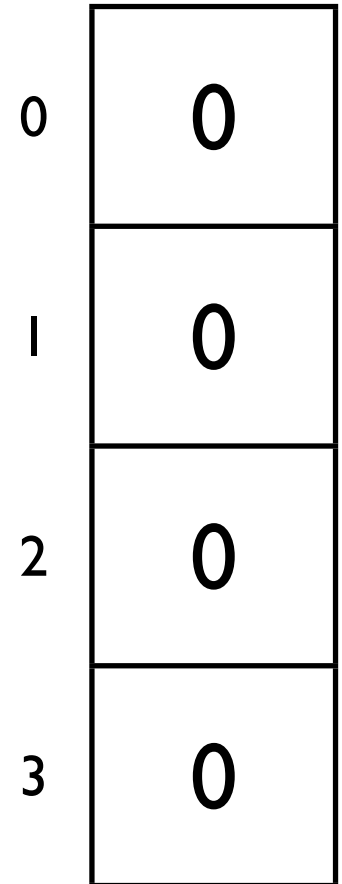
# Liste



- Definere en liste:
  - `hoydeAar = [50, 76, 87, 96]`
  - `hoydeAar = []`

# Liste

- Definere en liste:
  - `hoydeAar = [50, 76, 87, 96]`
  - `hoydeAar = []`
  - `hoydeAar = [0] * 4`



# Liste

- Definere en liste:
  - `hoydeAar = [50, 76, 87, 96]`
  - `hoydeAar = []`
  - `hoydeAar = [0] * 4`
- Sette en enkeltverdi:
  - `hoydeAar[0] = 5`

|   |   |
|---|---|
| 0 | 5 |
| 1 | 0 |
| 2 | 0 |
| 3 | 0 |

# Liste

- Definere en liste:
  - `hoydeAar = [50, 76, 87, 96]`
  - `hoydeAar = []`
  - `hoydeAar = [0] * 4`
- Sette en enkeltverdi:
  - `hoydeAar[0] = 5`
  - `hoydeAar[2] = 8`

|   |   |
|---|---|
| 0 | 5 |
| 1 | 0 |
| 2 | 8 |
| 3 | 0 |

# Liste

- Definere en liste:
  - `hoydeAar = [50, 76, 87, 96]`
  - `hoydeAar = []`
  - `hoydeAar = [0] * 4`
- Sette en enkeltverdi:
  - `hoydeAar[0] = 5`
  - `hoydeAar[2] = 8`
- Bruke enkeltverdi
  - `print(hoydeAar[2])`

|   |   |
|---|---|
| 0 | 5 |
| 1 | 0 |
| 2 | 8 |
| 3 | 0 |



# Liste

- Definere en liste:
  - `hoydeAar = [50, 76, 87, 96]`
  - `hoydeAar = []`
  - `hoydeAar = [0] * 4`
- Sette en enkeltverdi:
  - `hoydeAar[0] = 5`
  - `hoydeAar[2] = 8`
- Bruke enkeltverdi
  - `print(hoydeAar[2])`

|   |   |
|---|---|
| 0 | 5 |
| 1 | 0 |
| 2 | 8 |
| 3 | 0 |

Utvide en liste

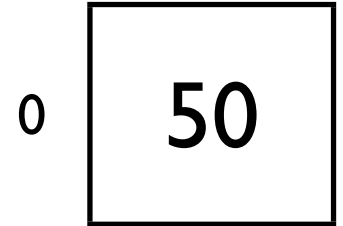
# Utvide en liste



- Først definere en tom liste
  - `hoydeAar = []`

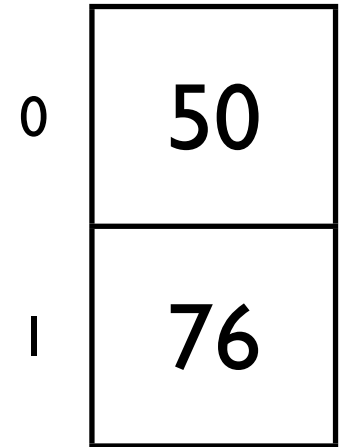
# Utvide en liste

- Først definere en tom liste
  - `hoydeAar = []`
- Utvide med en enkeltverdi:
  - `hoydeAar.append(50)`



# Utvide en liste

- Først definere en tom liste
  - `hoydeAar = []`
- Utvide med en enkeltverdi:
  - `hoydeAar.append(50)`
  - `hoydeAar.append(76)`



# Utvide en liste

- Først definere en tom liste
  - `hoydeAar = []`
- Utvide med en enkeltverdi:
  - `hoydeAar.append(50)`
  - `hoydeAar.append(76)`
- Konkatenerere lister
  - `print( [50,76] + [87,96] )`

|   |    |
|---|----|
| 0 | 50 |
| 1 | 76 |
| 2 | 87 |
| 3 | 96 |

# Utvide en liste

- Først definere en tom liste
  - `hoydeAar = []`
- Utvide med en enkeltverdi:
  - `hoydeAar.append(50)`
  - `hoydeAar.append(76)`
- Konkatenerere lister
  - `print( [50,76] + [87,96] )`
  - `hoydeAar = hoydeAar + [87,96]`

|   |    |
|---|----|
| 0 | 50 |
| 1 | 76 |
| 2 | 87 |
| 3 | 96 |

# Liste - funksjonalitet



# Liste - funksjonalitet

- Kan inneholde alle typer verdier

# Liste - funksjonalitet

- Kan inneholde alle typer verdier
  - `min_liste = [1.5, 2.9, 1.0]`

# Liste - funksjonalitet

- Kan inneholde alle typer verdier
  - `min_liste = [1.5, 2.9, 1.0]`
  - `min_liste = ["Oslo", "Bergen"]`

# Liste - funksjonalitet

- Kan inneholde alle typer verdier
  - `min_liste = [1.5, 2.9, 1.0]`
  - `min_liste = ["Oslo", "Bergen"]`
- Lengde av liste:

# Liste - funksjonalitet

- Kan inneholde alle typer verdier
  - `min_liste = [1.5, 2.9, 1.0]`
  - `min_liste = ["Oslo", "Bergen"]`
- Lengde av liste:
  - `len(min_liste)` **2**

# Liste - funksjonalitet

- Kan inneholde alle typer verdier
  - `min_liste = [1.5, 2.9, 1.0]`
  - `min_liste = ["Oslo", "Bergen"]`
- Lengde av liste:
  - `len(min_liste)` **2**
- Sjekke om en verdi finnes:

# Liste - funksjonalitet

- Kan inneholde alle typer verdier
  - `min_liste = [1.5, 2.9, 1.0]`
  - `min_liste = ["Oslo", "Bergen"]`
- Lengde av liste:
  - `len(min_liste)` **2**
- Sjekke om en verdi finnes:
  - `"Bergen" in min_liste` **True**

# Liste - funksjonalitet

- Kan inneholde alle typer verdier
  - `min_liste = [1.5, 2.9, 1.0]`
  - `min_liste = ["Oslo", "Bergen"]`
- Lengde av liste:
  - `len(min_liste)` **2**
- Sjekke om en verdi finnes:
  - `"Bergen" in min_liste` **True**
  - `"Trondheim" in min_liste` **False**



Noen tjenester  
som lister tilbyr

# Noen tjenester som lister tilbyr

- Telle

# Noen tjenester som lister tilbyr

- Telle
  - `liste = [1945, 1814, 1905, 1945]`  
`print( liste.count(1945) )` # 2

# Noen tjenester som lister tilbyr

- Telle
  - `liste = [1945, 1814, 1905, 1945]`  
`print( liste.count(1945) )` # 2
- Sortere

# Noen tjenester som lister tilbyr

- Telle

- `liste = [1945, 1814, 1905, 1945]`  
`print( liste.count(1945) )` # 2

- Sortere

- `liste.sort()`  
`print( liste )` # [1814, 1905, 1945, 1945]

En streng er en spesiell  
type liste

# En streng er en spesiell type liste

- tekst = "kamel"

# En streng er en spesiell type liste

- tekst = "kamel"
- sorted(tekst) # "aeklm"



# En streng er en spesiell type liste

- `tekst = "kamel"`
- `sorted(tekst) # "aeklm"`
- `tekst.count("m") # 1`

# En streng er en spesiell type liste

- `tekst = "kamel"`
- `sorted(tekst) # "aeklm"`
- `tekst.count("m") # 1`
- `tekst.append("a") #streng er immutable - kan ikke endres`

# En streng er en spesiell type liste

- `tekst = "kamel"`
- `sorted(tekst) # "aeklm"`
- `tekst.count("m") # 1`
- `tekst.append("a")` #streng er *immutable* - kan ikke endres
- `print( list(tekst) ) # ["k","a","m","e","l"]` (vanlig liste av bokstaver)

# Hva skrives ut?

```
vest = ["Halla", "Bergen"]
midt = ["Trondheim"]
print(vest + midt)
```

```
nord = ["Alta", "Kautokeino"]
vest = nord + vest
print(vest)
```

```
nord.append("Narvik")
print(nord)
```

```
lengde = len(vest+nord)
print(lengde)
```

# Hva skrives ut?

```
vest = ["Halla", "Bergen"]
midt = ["Trondheim"]
print(vest + midt) ['Halla', 'Bergen', 'Trondheim']
```

```
nord = ["Alta", "Kautokeino"]
vest = nord + vest
print(vest)
```

```
nord.append("Narvik")
print(nord)
```

```
lengde = len(vest+nord)
print(lengde)
```

# Hva skrives ut?

```
vest = ["Halla", "Bergen"]
midt = ["Trondheim"]
print(vest + midt) ['Halla', 'Bergen', 'Trondheim']
```

```
nord = ["Alta", "Kautokeino"]
vest = nord + vest
print(vest) ['Alta', 'Kautokeino', 'Halla', 'Bergen']
```

```
nord.append("Narvik")
print(nord)
```

```
lengde = len(vest+nord)
print(lengde)
```

# Hva skrives ut?

```
vest = ["Halla", "Bergen"]
midt = ["Trondheim"]
print(vest + midt) ['Halla', 'Bergen', 'Trondheim']
```

```
nord = ["Alta", "Kautokeino"]
vest = nord + vest
print(vest) ['Alta', 'Kautokeino', 'Halla', 'Bergen']
```

```
nord.append("Narvik")
print(nord) ['Alta', 'Kautokeino', 'Narvik']
```

```
lengde = len(vest+nord)
print(lengde)
```

# Hva skrives ut?

```
vest = ["Halla", "Bergen"]
midt = ["Trondheim"]
print(vest + midt) ['Halla', 'Bergen', 'Trondheim']
```

```
nord = ["Alta", "Kautokeino"]
vest = nord + vest
print(vest) ['Alta', 'Kautokeino', 'Halla', 'Bergen']
```

```
nord.append("Narvik")
print(nord) ['Alta', 'Kautokeino', 'Narvik']
```

```
lengde = len(vest+nord)
print(lengde) 7
```



En liten oppgave

# En liten oppgave

- A. Lag en liste med fem terningkast (tall fra 1 til 6) som du leser inn fra tastaturet (input)

# En liten oppgave

A. Lag en liste med fem terningkast (tall fra 1 til 6) som du leser inn fra tastaturet (input)

- Prøv selv med blyant og papir!

# En liten oppgave

A. Lag en liste med fem terningkast (tall fra 1 til 6) som du leser inn fra tastaturet (input)

- Prøv selv med blyant og papir!
- Etterpå diskuter med nabo

En liten oppgave

# En liten oppgave

- A. Lag en liste med fem terningkast (tall fra 1 til 6) som du leser inn fra tastaturet (input)

# En liten oppgave

- A. Lag en liste med fem terningkast (tall fra 1 til 6) som du leser inn fra tastaturet (input)
- B. Brukeren spiller yatsy og vil bruke sitt kast som firere - hvor mange poeng får brukeren (hun får fire poeng for hver firer hun har)

# En liten oppgave

- A. Lag en liste med fem terningkast (tall fra 1 til 6) som du leser inn fra tastaturet (input)
  - B. Brukeren spiller yatsy og vil bruke sitt kast som firere - hvor mange poeng får brukeren (hun får fire poeng for hver firer hun har)
- 
- Prøv selv med blyant og papir!



# En liten oppgave

- A. Lag en liste med fem terningkast (tall fra 1 til 6) som du leser inn fra tastaturet (input)
  - B. Brukeren spiller yatsy og vil bruke sitt kast som firere - hvor mange poeng får brukeren (hun får fire poeng for hver firer hun har)
- 
- Prøv selv med blyant og papir!
  - Etterpå diskuter med nabo

# Løsning på A

(lage liste med fem terningkast)

# Løsning på A

(lage liste med fem terningkast)

- {firere1.py}

# Løsning på B

(telle poeng for firere)

# Løsning på B

(telle poeng for firere)

- {firere2.py}

Flere utfordringer?

# Flere utfordringer?

- Sjekk ut Fredagspython: et ekstra tilbud for kreativ og utforskende programmering

# Flere utfordringer?

- Sjekk ut Fredagspython: et ekstra tilbud for kreativ og utforskende programmering
- Få innspill! Prøv deg frem! Utvid horisonten!



# Flere utfordringer?

- Sjekk ut Fredagspython: et ekstra tilbud for kreativ og utforskende programmering
- Få innspill! Prøv deg frem! Utvid horisonten!
- Fredager 1615 - 1800 på Assembler