1 Sekvenser

Sekvenser er en samling av elementer i en ordnet rekkefølge, som er grunnleggende datastrukturer i programmering for effektiv datahåndtering. I tillegg i Python, tilbyr sekvenser en rekke innebygde operasjoner som enkelt gir oss grunnleggende funksjonaliteter.

1.1 Liste

I python, er lister en allsidig datastruktur for lagring og manipulasjon av elementer som kan legges til, fjernes, oppdateres og manipuleres enkelt.

- En liste er en ordnet, foranderlig sekvens
- Elementene i en liste kan være av forskjellige datatyper.
- Vi bruker syntaksen `[element1, element2, ...]` for å opprette en liste.
- En liste har datatypen `list` i Python.

[1]:
```
# En tom liste
tom_liste = []
# En liste med forskjellige typer
min_liste = [2024, 0.1, True, None, "DIGI"]
print(min_liste)

print("Typen til en liste er: ", {type(min_liste)})
```

```
[2024, 0.1, True, None, 'DIGI']
Typen til en liste er:  {<class 'list'>}
```

1.1.1 Grunnleggende operasjoner på sekvenser

- Vi kan få tilgang til elementene i en sekvens ved å referere til deres indeksnummer, f. eks. `min_liste[3]`.
- Indeksen til det første elementet er `0`.
- Vi kan beregne lengden på en sekvens ved å bruke funksjonen `len()`

Kommentar om utførelsen: Det oppstår en feil!

[2]:
```
min_liste = [2024, 0.1, True, None, "DIGI"]

lengden = len(min_liste)
print(f"Listen har {lengden} elementer")

# Indeksering begynner med indeks 0, ikke 1
print(f"Det første elementet i listen min er: {min_liste[0]}")
# Det er femte (siste) elementet
print(f"Det siste elementet i listen min er:  {min_liste[lengden-1]}")
# /!\ Det utløser en feilmelding /!\: Det er ingen element ved indeks 5
print(f"Det er ingen element ved indeks 5:    {min_liste[lengden]}")
```

```
Listen har 5 elementer
Det første elementet i listen min er: 2024
Det siste elementet i listen min er:  DIGI
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
```

```
Cell In[2], line 11
      9 print(f"Det siste elementet i listen min er:  {min_liste[lengden-1]}")
     10 # /!\ Det utløser en feilmelding /!\: Det er ingen element ved indeks 5
---> 11 print(f"Det er ingen element ved indeks 5:    {min_liste[lengden]}")

IndexError: list index out of range
```

1.1.2 Liste: en foranderlig sekvens, Streng: en uforanderlig sekvens

Vi kan oppdatere en liste men vi kan ikke oppdatere en streng.

Kommentar om utførelsen: Det oppstår en feil!

[3]:
```
min_liste = [2024, 0.1, "DIGI", True, None]
min_streng = "Hei hei, verden!"

# Syntaksen f"{uttrykk = }" skiver både uttrykket og evalueringen (resultatet)
# av uttrykket
print(f"{min_liste = }")
print(f"{min_streng = }")

# Vi kan endre verdiene i listen
min_liste[1] = "ny_verdi"
print(min_liste)
# Men vi kan ikke endre verdiene i streng
# /!\ Det utløser en feilmelding /!\: en streng er uforanderlig
min_streng[1] = "e"
print(min_streng)
```

```
min_liste = [2024, 0.1, 'DIGI', True, None]
min_streng = 'Hei hei, verden!'
[2024, 'ny_verdi', 'DIGI', True, None]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[3], line 14
     11 print(min_liste)
     12 # Men vi kan ikke endre verdiene i streng
     13 # /!\ Det utløser en feilmelding /!\: en streng er uforanderlig
---> 14 min_streng[1] = "e"
     15 print(min_streng)

TypeError: 'str' object does not support item assignment
```

1.1.3 Flere metoder på sekvenser (f.eks. lister og strenger)

Lister og strenger er 2 undertyper av sekvenser. Undertypene av sekvenser har tilgang til sekvens metoder. Vi kan finne alle felles metoder for sekvenser i python her: https://docs.python.org/3/library/stdtypes.html?highlight=list#common-sequence-operations.

Det forventes ikke at dere lærer alle disse metodene utenat! I cellen nedenfor gir vi dere bare en oversikt over hva som er mulig!

[4]:
```
min_liste = [2024, 0.1, "DIGI", True, None]
min_streng = "Hei hei, verden!"
print(f"{min_liste = }")
print(f"{min_streng = }")

print("\n --- Lengden på sekvensen ---")
# "len()": hvor mange elementer finnes det i en sekvens
print(f"{len(min_liste) = }")
print(f"{len(min_streng) = }")

print("\n --- Konkatenering av sekvenser --- ")
# " + ": konkatenering av 2 sekvenser
l = min_liste + ["Hei", 0]
print(f'min_liste + ["Hei", 0]: {l}')
s = min_streng + ":)"
print(f'min_streng + ":)": {s}')

print("\n --- Sjekk om noe er i sekvensen --- ")
# " in ": Sjekk om noe er i sekvensen
print(f"{None in min_liste = }")
print(f"{'h' in min_streng = }")   # 'H' er i strengen men ikke 'h'!

print("\n --- Indeksering, segment --- ")
# Trekk ut flere elementer fra en sekvens
# Syntaksen: [start:]
print(f"{min_streng[5:] = }")
# Syntaksen: [start:slutt]
print(f"{min_liste[1:4] = }")
print(f"{min_streng[5:11] = }")
# Syntaksen: [start:slutt:steg]
print(f"{min_streng[1:11:2] = }")

# "index()": finn plassering til noe i sekvensen
print(f"{min_liste.index(True) = }")
# Merk at kun indeksen til den første 'e' returneres
print(f"{min_streng.index('e') = }")

print("\n --- min og max --- ")
print(f"{min([12, -2, 0.1]) = }")
print(f"{max(min_streng) = }")
```

```
min_liste = [2024, 0.1, 'DIGI', True, None]
min_streng = 'Hei hei, verden!'

 --- Lengden på sekvensen ---
len(min_liste) = 5
len(min_streng) = 16

 --- Konkatenering av sekvenser ---
min_liste + ["Hei", 0]: [2024, 0.1, 'DIGI', True, None, 'Hei', 0]
min_streng + ":)": Hei hei, verden!:)
```

```
 --- Sjekk om noe er i sekvensen ---
None in min_liste = True
'h' in min_streng = True

 --- Indeksering, segment ---
min_streng[5:] = 'ei, verden!'
min_liste[1:4] = [0.1, 'DIGI', True]
min_streng[5:11] = 'ei, ve'
min_streng[1:11:2] = 'e e,v'
min_liste.index(True) = 3
min_streng.index('e') = 1

 --- min og max ---
min([12, -2, 0.1]) = -2
max(min_streng) = 'v'
```

Metoder på sekvenser: vanlige feil **Det forventes ikke at dere kjenner disse detaljene utenat! I cellen nedenfor prøver vi:**

- å gjøre dere mer kjent til å lese feilmeldinger
- å gjøre dere mer kjent til å lese dokumentasjon / søke på internett hvis de oppstår en feil

Kommentar om utførelsen: Det oppstår en feil!

[5]:
```
# /!\ Det utløser en feilmelding /!\:
# Hvis elementet ikke er i listen utløser index() en feil!
print(f"{min_streng.index('i') = }")

# /!\ Det utløser en feilmelding /!\:
# en streng konkateneres bare med en streng og en list bare med en list!
print(f"{[0, 'Hallo'] + 'verden'}")

# /!\ Det utløser en feilmelding /!\: kan ikke sammenligne streng og tall
print(f"{min(min_liste) = }")
```

```
min_streng.index('i') = 2
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[5], line 7
      3 print(f"{min_streng.index('i') = }")
      5 # /!\ Det utløser en feilmelding /!\:
      6 # en streng konkateneres bare med en streng og en list bare med en list!
----> 7 print(f"{[0, 'Hallo'] + 'verden'}")
      9 # /!\ Det utløser en feilmelding /!\: kan ikke sammenligne streng og tall
     10 print(f"{min(min_liste) = }")

TypeError: can only concatenate list (not "str") to list
```

1.1.4 Metoder på strenger

En streng har også tilgang til spesielle metoder på strenger. Vi kan finne alle streng metoder i python her: https://docs.python.org/3/library/stdtypes.html#string-methods.

Det forventes ikke at dere lærer alle disse metodene utenat! I cellen nedenfor gir vi dere bare en oversikt over hva som er mulig!

[6]:
```
min_streng = "Hei hei, verden!"
vennene = ["Nora", "Odin", "Morten" ]
print(f"{min_streng = }")
print(f"{vennene = }")

print("\n --- Erstatte en understreng i strengen ---")
# "replace()": Erstatte alle 'e' med en 'i'
print(f"{min_streng.replace('e', 'i') = }")
# Ingenting skjer fordi bokstaven ikke var i strengen
print(f"{min_streng.replace('w', 'A') = }")
# Det fungerer også med sekvenser av tegn
print(f"{min_streng.replace('ver', 'klo') = }")

print("\n --- Erstatte en understreng i strengen ---")
# "split()": dele strengen inn i en liste med strenger
print(f"{min_streng.split(' ') = }")
print(f"{min_streng.split(',') = }")

print("\n --- Sjekke om strenger begynner/slutter med ---")
print(f"{min_streng.startswith('hei') = }")
print(f"{min_streng.endswith('!') = }")

print("\n --- Skrive med bare store/små bokstaver ---")
# "upper()": Skrive med bare store bokstaver
print(f"{min_streng.upper() = }")
# "lower()": Skrive med bare små bokstaver
print(f"{min_streng.lower() = }")
# Man kan kombinere flere metoder
print(f"{min_streng.lower().startswith('hei') = }")

print("\n --- Slår en list sammen til en streng ---")
# Slå sammen hvert element av vennene men en "og " innimellom
print(f"{' og '.join(vennene) = }")
```

```
min_streng = 'Hei hei, verden!'
vennene = ['Nora', 'Odin', 'Morten']

 --- Erstatte en understreng i strengen ---
min_streng.replace('e', 'i') = 'Hii hii, virdin!'
min_streng.replace('w', 'A') = 'Hei hei, verden!'
min_streng.replace('ver', 'klo') = 'Hei hei, kloden!'

 --- Erstatte en understreng i strengen ---
min_streng.split(' ') = ['Hei', 'hei,', 'verden!']
min_streng.split(',') = ['Hei hei', ' verden!']
```

```
 --- Sjekke om strenger begynner/slutter med ---
min_streng.startswith('hei') = False
min_streng.endswith('!') = True

 --- Skrive med bare store/små bokstaver ---
min_streng.upper() = 'HEI HEI, VERDEN!'
min_streng.lower() = 'hei hei, verden!'
min_streng.lower().startswith('hei') = True

 --- Slår en list sammen til en streng ---
' og '.join(vennene) = 'Nora og Odin og Morten'
```

1.1.5 Metoder på *foranderlige* sekvenser

En liste har også tilgang til alle metodene på *foranderlige* sekvenser. Vi kan finne alle streng metoder i python her: https://docs.python.org/3/library/stdtypes.html?highlight=list#mutable-sequence-types.

Det forventes ikke at dere lærer alle disse metodene utenat! I cellen nedenfor gir vi dere bare en oversikt over hva som er mulig!

[7]:
```python
min_liste = [2024, '!', "DIGI", True, None]
print(f"{min_liste = }")

print("\n --- Legge til et element på slutten (på stedet) ---")
# Vær forsiktig! "append" metoden er en "på stedet metode (in-place method)"
# Det skriver ut None, det betyr at returverdien til "append" metoden er "None"
# Det er fordi det er en "på stedet metode" som ikke returner listen, men endrer
# listen direkte
print(f"{min_liste.append(42) = }")
# Det skriver ut listen med det nye elementet
print(f"{min_liste = }")

print("\n --- Legge til et element på en gitt indeks (på stedet) ---")
# Legg "42" på indeks 1 (dvs den andre posisjonen)
# Vær forsiktig! "insert" er også en "på stedet metode "
print(f"{min_liste.insert(1, 42) = }")
print(f"{min_liste = }")

print("\n --- Fjern et gitt element fra sekvensen (på stedet) ---")
# Merk at kun den første 42 ble fjernet!
# remove er også en "på stedet metode"
print(f"{min_liste.remove(42) = }")
print(f"{min_liste = }")

print("\n --- Fjern elementet som ligger på en gitt indeks (på stedet) ---")
# pop er en på stedet metode (fordi endringene skjer direkte på listen,
# men den returnerer fortsatt noe: elementet som ble fjernet)
print(f"{min_liste.pop(2) = }")
print(f"{min_liste = }")
```

```
min_liste = [2024, '!', 'DIGI', True, None]

 --- Legge til et element på slutten (på stedet) ---
min_liste.append(42) = None
min_liste = [2024, '!', 'DIGI', True, None, 42]

 --- Legge til et element på en gitt indeks (på stedet) ---
min_liste.insert(1, 42) = None
min_liste = [2024, 42, '!', 'DIGI', True, None, 42]

 --- Fjern et gitt element fra sekvensen (på stedet) ---
min_liste.remove(42) = None
min_liste = [2024, '!', 'DIGI', True, None, 42]

 --- Fjern elementet som ligger på en gitt indeks (på stedet) ---
min_liste.pop(2) = 'DIGI'
min_liste = [2024, '!', True, None, 42]
```

Metoder på *foranderlige* sekvenser: vanlige feil **Det forventes ikke disse detaljene utenat! I cellen nedenfor prøver vi:** - å gjøre dere mer kjent til å lese feilmeldinger - å gjøre dere mer kjent til å lese dokumentasjon / søke på internett hvis de oppstår en feil

Kommentar om utførelsen: Det oppstår en feil!

[8]:
```
# /!\ Det utløser en feilmelding /!\:
# Hvis elementet ikke er i listen utløser index() en feil!
print(f"{min_liste.remove(100) = }")
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[8], line 3
      1 # /!\ Det utløser en feilmelding /!\:
      2 # Hvis elementet ikke er i listen utløser index() en feil!
----> 3 print(f"{min_liste.remove(100) = }")

ValueError: list.remove(x): x not in list
```

1.1.6 Metoder på lister: spesielle metoder på lister

En liste har i tillegg metoder spesielt for lister. Se https://docs.python.org/3/library/stdtypes.html?highlight=list#list.

Det forventes ikke at dere lærer alle denne metoden utenat! I cellen nedenfor gir vi dere bare en oversikt over hva som er mulig!

[9]:
```
karakterene  = [1, 8, 3, 0, 10, 1]
vennene = ["Nora", "Odin", "Morten" ]
print(f"{karakterene = }")
print(f"{vennene = }")

print("\n --- Sortere en list (på stedet) ---")
```

```
print(f"{karakterene.sort() = }")
print(f"{karakterene = }")

# sort metoden har en parameter "reverse" som bestemmer hvorvidt
# listen må sorteres i økende rekkefølge eller motsatt vei.
print(f"{vennene.sort(reverse=True) = }")
print(f"{vennene = }")
```

```
karakterene = [1, 8, 3, 0, 10, 1]
vennene = ['Nora', 'Odin', 'Morten']

 --- Sortere en list (på stedet) ---
karakterene.sort() = None
karakterene = [0, 1, 1, 3, 8, 10]
vennene.sort(reverse=True) = None
vennene = ['Odin', 'Nora', 'Morten']
```

Metoder på lister: vanlige feil Det forventes ikke disse detaljene utenat! **I cellen nedenfor prøver vi:** - å gjøre dere mer kjent til å lese feilmeldinger - å gjøre dere mer kjent til å lese dokumentasjon / søke på internett hvis de oppstår en feil

Kommentar om utførelsen: Det oppstår en feil!

[10]:
```
min_liste = [2024, 0.1, "DIGI", True, None]
print(f"{min_liste = }")

# /!\ Det utløser en feilmelding /!\:
# Kan ikke sammenligne str med list og med bool!
min_liste.sort()
```

```
min_liste = [2024, 0.1, 'DIGI', True, None]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[10], line 6
      2 print(f"{min_liste = }")
      4 # /!\ Det utløser en feilmelding /!\:
      5 # Kan ikke sammenligne str med list og med bool!
----> 6 min_liste.sort()

TypeError: '<' not supported between instances of 'str' and 'float'
```

1.2 Enda et eksempel: Liste av pakkene som må flyttes

[11]:
```
# Legg noe til på slutten av listen
pakkene = [1, 8, 3, 0, 10, 1]
print(pakkene)
siste_pakke = int(input("Hva er volumet til den neste pakken din?"))
pakkene.append(siste_pakke)
print("En pakke var lagt til på slutten", pakkene)
```

```python
# Sett noe inn i listen på en gitt indeks
ny_pakke = int(input("Angi volumet som skall settes inn på tredjeplassen."))
# Husk at tredjeplassen tilsvarer indeksen 2
pakkene.insert(2, ny_pakke)
print("En pakke var settes inn på tredjeplassen", pakkene)

# Fjern et element fra listen
pakkene.remove(ny_pakke)
print("Pakken på tredjeplassen ble fjernet", pakkene)

# Sortere en liste
pakkene.sort(reverse=False)
print("Listen ble sortert", pakkene)
```

```
[1, 8, 3, 0, 10, 1]
En pakke var lagt til på slutten [1, 8, 3, 0, 10, 1, 32]
En pakke var settes inn på tredjeplassen [1, 8, 58, 3, 0, 10, 1, 32]
Pakken på tredjeplassen ble fjernet [1, 8, 3, 0, 10, 1, 32]
Listen ble sortert [0, 1, 1, 3, 8, 10, 32]
```