

# Modul 2 - Del 2 - Hovedpoeng

---

## Hovedpoeng

- Det er viktig å lese dokumentasjonen nøye for funksjonene/metodene vi bruker.
- Ikke vær redd for å søke informasjon på internett. Sørg bare for at du bruker riktige ord for å søke effektivt.
- Man trenger ikke å lære alt om alle de metode/bibliotekene som eksisterer, men i programmering er det viktig å lære å finne informasjon raskt om en gitt funksjon/bibliotek.
- Python har [en rekke innebygde funksjoner](#) som gir grunnleggende funksjonalitet og operasjoner
- Det er [foranderlige sekvenstyper](#) i Python (`list`) og uforanderlige sekvenstyper (`str`). I tillegg deler alle sekvenstyper [felles metoder](#).
- En spesifikk type kan ha ekstra metoder. For eksempel [lister](#) og [strenger](#).
- Husk hvilken type variablene du arbeider med har. Resultatet av en gitt operasjon kan avhenge av variabeltypen, og de tilgjengelige metodene avhenger også av variabeltypen.

## Jukseark

Påminnelse: det forventes ikke at dere kjenner det utenat!

### Datatyper

Man kan sjekke typen til et objekt med funksjonen `type()`.

Navn (norsk)	Navn (engelsk)	Eksempel	Konvertere til
Liste	List	<code>["Nora", "Odin", "Morten"]</code>	<code>list()</code>

`None` er et spesielt objekt som representerer fraværet av en verdi eller en null-verdi. `None` er et unikt objekt av typen `NoneType`.

### Lister

```
tom_liste = []
land_liste = ["Norge", "Danmark", "Sverige"]
karakter_liste = [4, 3, 2, 8, 3]
rotete_liste = [2024, 0.1, "DIGI", True, None]

# Indeksen til det første elementet er `0`
print(land_liste[1])           # "Danmark"
```

### Vanlige innebygde funksjoner og metoder

#### Innebygde funksjoner

#### Sekvens metoder

```
liste = [2024, 0.1, "DIGI", True, None]
streng = "Hei hei, verden!"
```

Indeksering syntaks: `liste[start:slutt:steg]` med standardverdier (dvs. om utelatt):

- `start = 0`
- `slutt = -1`
- `steg = 1`

Metode / Operasjon	Resultat	Beskrivelse
<code>liste[1]</code>	<code>0.1</code>	Hent ut det som ligger på indeks <code>1</code>
<code>streng[-1]</code>	<code>"!"</code>	Hent ut det som ligger <code>-1</code> fra slutten av sekvensen
<code>liste[-2]</code>	<code>True</code>	Hent ut det som ligger <code>-2</code> fra slutten av sekvensen
<code>streng[:3]</code>	<code>Hei</code>	Hent ut de som ligger fra indeksen 0 til (ikke inkludert) indeksen 3
<code>liste[1:4]</code>	<code>[0.1, 'DIGI', True]</code>	Hent ut de som ligger fra indeksen 1 til (ikke inkludert) indeksen 4
<code>streng[1:11:2]</code>	<code>'e e,v'</code>	Hent ut hver annen element, startende fra indeksen 1 til (ikke inkludert) indeksen 11
<code>len(liste)</code>	<code>5</code>	Lengden på sekvensen
<code>len(streng)</code>	<code>16</code>	Lengden på sekvensen
<code>["DIGI", True] + [2024, 0.1]</code>	<code>["DIGI", True, 2024, 0.1]</code>	Slå sammen sekvenser
<code>'verd' in streng</code>	<code>True</code>	Sjekk om noe er i sekvensen
<code>None not in liste</code>	<code>False</code>	Sjekk om noe ikke er i sekvensen
<code>liste.index("DIGI")</code>	<code>2</code>	Finn plassering til noe i sekvensen

## Foranderlige sekvens metoder

Disse metodene er "in-place"-metoder. Det betyr at de ikke returnerer sekvensen, og i stedet oppdaterer de sekvensen på stedet. Dette er mulig fordi vi her jobber med foranderlige sekvenser.

Metode / Operasjon	Resultat	Beskrivelse
<code>liste[1] = "!"</code>	<code>[2024, '!', 'DIGI', True, None]</code>	Sett det som ligger på indeks <code>1</code> til <code>"!"</code>
<code>liste.append(42)</code>	<code>[2024, '!', 'DIGI', True, None, 42]</code>	Legg noe til på slutten av sekvensen
<code>liste.insert(1, 42)</code>	<code>[2024, 42, '!', 'DIGI', True, None, 42]</code>	Sett noe inn i sekvensen på en gitt indeks

Metode / Operasjon	Resultat	Beskrivelse
<code>liste.remove(42)</code>	<code>[2024, '!', 'DIGI', True, None, 42]</code>	Fjern et gitt element fra sekvensen.
<code>liste.pop(2)</code>	<code>[2024, '!', True, None, 42]</code>	Fjern elementet som ligger på en gitt indeks

#### Liste metoder

```

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

```

Metode / Operasjon	Resultat	Beskrivelse
<code>karakterene.sort()</code>	<code>[0, 1, 1, 3, 8, 10]</code>	Sorter listen (om mulig!)
<code>vennene.sort(reverse=True)</code>	<code>['Odin', 'Nora', 'Morten']</code>	Sorter listen (om mulig!) motsatt vei

#### Streng metoder

Disse metodene er ikke "in-place"-metoder. De returnerer en ny sekvens og lar den opprinnelige sekvensen være intakt. Dette skyldes at vi her jobber med uforanderlige sekvenser.

Metode / Operasjon	Resultat	Beskrivelse
<code>streng.upper()</code>	<code>"HEI HEI, VERDEN!"</code>	Gjør alle bokstaver store
<code>streng.lower()</code>	<code>"hei hei, verden!"</code>	Gjør alle bokstaver små
<code>streng.startswith('hei')</code>	<code>False</code>	Sjekke om strengen begynner med
<code>streng.endswith('!')</code>	<code>True</code>	Sjekke om strengen slutter med
<code>streng.split(",")</code>	<code>['Hei hei', ' verden!']</code>	Deler strengen i en liste
<code>streng.split(" ")</code>	<code>['Hei', 'hei,', 'verden!']</code>	Deler strengen i en liste
<code>streng.replace('e', 'i')</code>	<code>Hii hii, virdin!</code>	Erstatter en understreng i strengen
<code>streng.replace('ver', 'klo')</code>	<code>Hei hei, kloden!</code>	Erstatter en understreng i strengen
<code>" og ".join(vennene)</code>	<code>"Nora og Odin og Morten"</code>	Slår en list sammen til en streng