Uke 3, time 1

Onsdag 3. september

Introduksjon til Python

Konseptforståelse

Objekter

Samlinger

- Lister
- Mengder
- Ordbøker
- Nøstede samlinger

Objekter

Objekter er instanser av klasser

Når vi oppretter et heltall, flyttall, en string (tekst) osv. så er det objekter av de innebygde klassene `int`, `float`, `str`.

For nå skal vi se på hva det er de innebygde klassene tilbyr av tjenester

Tjenester som en klasse tilbyr kaller vi også for metoder, men vi har også konstruktører og instans– og klassevariabler

klassen str

I klassen for strenger har vi mange tjenester som er nyttig

Her er noen eksempler:

`lower()`, `upper()`, `casefold()`, `capitalize()`

`count()`

`endswith()`, `removesuffix()`

Vi kan også bruke konstruktøren `str()` for å lage et nytt objekt av typen string, f.eks fra et heltall

Samlinger

Til nå har variabler inneholdt én verdi, men vi har mange gode muligheter i Python til å lagre flere verdier

Vi kaller disse for samlinger (Collections)

Nå skal vi se på lister (list), mengder (set), ordbøker (dictionary)

Lister

Lister er rett og slett en liste av verdier (objekter) med en bestemt rekkefølge

Hvert element har en indeks (som begynner på 0 for første element)



Lister

```
dyrene = ["sau", "hund", "katt", "giraff", "hest"]

print(dyrene[0])

dyrene.append("esel")

katte_dyr = ["tiger", "løve", "leopard"]

dyrene = dyrene + katte_dyr

print(len(dyrene))

if "tiger" in dyrene:
    print("Vi har en tiger blant dyrene")
```

Mengder

Mengder ligner på lister, men kan kun ha unike verdier. Det betyr at hvis man legger til en verdi med samme innhold, så vil den overskrives

Mengder har ikke en bestemt rekkefølge

Det er nyttig når man ønsker oversikt over noe som skal være unikt

Mengder

```
dyrene = {"sau", "sau", "katt", "giraff", "hest"}

print(dyrene)

print(len(dyrene))

dyrene.add("esel")
dyrene.add("sau")

katte_dyr = {"tiger", "løve", "leopard"}

dyrene = dyrene | katte_dyr

if "tiger" in dyrene:
    print("Vi har en tiger blant dyrene")
```

Mengder

```
dyrene = {"sau", "katt", "giraff", "hest"}

katte_dyr = {"katt", "tiger", "løve", "leopard"}

print(dyrene | katte_dyr)  # kombinerer alle
{'sau', 'tiger', 'hest', 'giraff', 'leopard', 'løve', 'katt'}

print(dyrene & katte_dyr)  # bare felles
{'katt'}

print(dyrene - katte_dyr)  # trekker fra første mengde
{'sau', 'hest', 'giraff'}

print(dyrene ^ katte_dyr)  # ulike, ikke i begge mengdene
{'sau', 'tiger', 'hest', 'giraff', 'leopard', 'løve'}
```

Ordbøker

Ordbøker er en type samling som kobler en nøkkelverdi til en annen verdi

Kalles også map, nøkkel/verdi (key-value)

Har i utgangspunktet heller ikke en rekkefølge, og nøkkelverdien er unik

"hund": "hundedyr"	"ulv": "hundedyr"	"katt": "kattedyr"	"tiger": "kattedyr"	"hest": "hestedyr"

"Norge": "Oslo"	"Finland": "Helsinki"	"Tyskland": "Berlin"	"Spania": "Madrid"	"Tyrkia": "Ankara"

Ordbøker

```
hundedyr = "hundedyr"
kattedyr = "kattedyr"

dyrene = {
    "hund": hundedyr,
    "ulv": hundedyr,
    "katt": kattedyr,
    "tiger": kattedyr,
    "hest": "hestedyr"
}

print(dyrene["tiger"])

dyrene["løve"] = kattedyr

if "ulv" in dyrene:
    print(Vi har en ulv som er et " + dyrene["ulv"])
```

Opprette tomme samlinger

```
# liste
min_liste = []
# eller
din_liste = list()

# mengde
min_mengde = set()

# ordbok
min_ordbok = {}
# eller
din_ordbok = dict()
```

Konvertere

```
# fra mengde til liste
min_liste = list(min_mengde)

# fra liste til mengde
min_mangde = set(min_liste)
```

Nøstede samlinger

Samlinger kan inneholde andre samlinger

Vi kan bl.a. lage "2D" matriser ved å la en liste inneholde andre lister

Ordbøker kan ha lister som en verdi til en nøkkel

Nøstede samlinger

```
mat_retter = {
    "hoved": ["biff", "kylling", "salat"],
    "saus": ["bernaise", "brun saus"],
    "dessert": ["iskrem", "banana-splitt"]
}
```

Trix

Oppgaver 03.xx

.