IN1000

Uke 2

Plan for dagen

- Datatyper og konvertering
- Uttrykk og evaluering
- Prosedyrer
- Kodeflyt
- Feilmeldinger
- Kodestil og indentering

Datatyper og konvertering

- De vanligste datatypene er:

String

```
"Dette er en string"
```

Integer



Float



Boolean





Datatyper og konvertering

- Noen ganger ønsker vi å konvertere til en annen datatype

```
str(4) == "4"
int("5") == 5
int(" 6 ") == 6 # mellomrommene ignoreres
float("3.14") == 3.14
```

Datatyper og konvertering

- Dette kan være fordi vi ønsker å ta imot en integer fra brukeren
- Eller når man ønsker å slå sammen en int og en string til én stor string

```
tall = input("Skriv inn et tall: ")
tall = int(tall)


tall = int(input("Skriv inn et tall: "))

print("Dette var tallet du skrev inn: " + str(tall))
```

Datatyper og konvertering

- Man kan sjekke hvilken datatype det er ved å bruke type()

```
variabel = 4.13

print(type(variabel)) #vil printe <class 'float'>
```

Uttrykk og evalueringer

- **Boolske verdier** er en datatype som enten er True eller False
- **Boolske uttrykk** er noe som evaluerer til en boolsk verdi (altså enten True eller False)
- F.eks.

```
boolean1 = 4 < 5

boolean2 = 10 < 1

boolean1 == True
boolean2 == False
```

Uttrykk og evalueringer

- Man kan slå sammen og moderere boolske uttrykk ved hjelp av **boolske operatorer**

- **and** – her må begge uttrykkene være sanne (evaluere til True)
- **or** – her må minst ett av uttrykkene være sanne (evaluere til True)
- **not** – denne evaluerer til det motsatte, slik at True blir False og omvendt

Oppgave!

- Hva evaluerer de boolske uttrykkene til?

```
1   #and
2   True and True
3   True and False
4
5   #or
6   True or True
7   True or False
8   False or False
9
10  #not
11  not(True)
12  not(False)
13
14  # Litt mer utfordrende
15  True and not(True)
16
17  True and not(False)
18
19  not(True) or not(False)
```

Et litt mer realistisk eksempel:

```
a = 4
b = 5

if a < b and a > 10:
    print("Denne evaluerte til True")
else:
    print("Denne evaluerte til False")

if a < b or a > 10:
    print("Denne evaluerte til True")
else:
    print("Denne evealuerte til False")
```

Sammenligninger:

Uttrykk	Betydning
a == b	a er lik b
a != b	a er ikke lik b
a > b	a er større enn b
a >= b	a er større eller lik b
a < b	a er mindre enn b
a <= b	a er mindre eller lik b

Oppgave!

Hva evaluerer uttrykkene til? (True vs. False)

```
5 < 5
5 <= 5
5 > 5
5 >= 5
5 == 5
5 != 5
```

Prosedyrer

- Hvorfor bruke bruke prosedyrer?
- Hvordan er kodeflyten?
- pythontutor.com
- Hvordan lage en prosedyre?

```
def prosedyre_navn():
    # det man vil at prosedyren skal gjøre
```

```
def si_hallo():
    print("Hallo!")

si_hallo()
si_hallo()
si_hallo()
```

Feilmeldinger

- Hva står det:
    1. filnavnet
    2. hvilken linje
    3. Type feil

```
File "HelloWorld.py", line 43
  print(Tallet er større enn 5)
                ^
SyntaxError: invalid syntax
```

```
Skriv inn et tall: w
Traceback (most recent call last):
  File "HelloWorld.py", line 37, in <module>
    tall = float(input("Skriv inn et tall: "))
ValueError: could not convert string to float: 'w'e
```

Oppgave – finn tre feil!

```
tall = int(input("Skriv inn et tall med to desimaler: "))

if tall < 5:
    print(Tallet er større enn 5)
else:
    print(Tallet er ikke større enn 5)
```

Kodestil og indentering

- Kommentarer
- Luft og lesbarhet
- Beskrivende navn – så korte som mulig
- snake_case, ikke camelCase

Kodestil og indentering

```
tall = 3

if tall > 20:
    print("Større")
else:
    print("Mindre")
```



```python
navn = "Nora"
alder = 16
if alder >= 18:
    print(navn, "er myndig.")
else:
    print(navn, "er ikke myndig.")
```

```python
n = "Nora"
a = 16
if a >= 18:
    print(n, "er myndig.")
else:
    print(n, "er ikke myndig.")
```

```python
NAVN = "Nora"
ALDER = 16
if ALDER >= 18:
    print(NAVN, "er myndig")
else:
    print(NAVN, "er ikke myndig")
```

```python
navn="Nora"
alder=16
if alder>=18:
print(navn,"er myndig.")
else:
    print(navn,"er ikke myndig")
```