



IN1000

Gruppe 16



Plan for timen

- Opprop
- Repetisjon
- Samkoding



Læringsmål uke 2

- Forstå hvordan én enkelt linje utføres:
 - Datatyper, evaluering av uttrykk og funksjoner
- Ha god forståelse av variabler
- Forstå og kunne bruke enkle prosedyrer uten parametre
- Forstå hvordan et helt program utføres -
 - kodeflyt fra linje til linje, inkludert for beslutninger og prosedyrer



Repetisjon: enkel input fra keyboard

- Syntaks: `navn = input()`
- Dersom man vil gi en utskrift før input:
 - `navn = input("Hva heter du?")`
- Det som skrives som input (frem til enter trykkes) lagres i variabelen `navn`.
- Dersom du ønsker at inputen fra bruker skal skrives inn på ny linje kan man legge inn `\n = newline` (eneste nye denne uken når det gjelder input)
 - `navn = input("Hva heter du?\n")`



Feilmeldinger, repetisjon

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

Unntak/exceptions

```
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'
```



Feilmeldinger, repetisjon

```
Traceback (most recent call last):
```

```
File "C:\Users\johan\OneDrive\Desktop\IN1000_gruppe\print.py", line 9, in <module>  
    print(naavn)
```

```
NameError: name 'naavn' is not defined
```

Noen feil gir ikke feilmeldinger, for eksempel hvis du som programmerer har tenkt feil, så kan dette gi logiske feil i programmet.

Eks. skal man regne ut gjennomsnittstemperaturen for ett år.

```
AntallMnd = 13  
snitt = Total / AntallMnd
```

> Dette vil gi feil fordi vi har feil antall mnd (det er 12 mnd i ett år).



Kommentarer

"""

komentar over

flere linjer

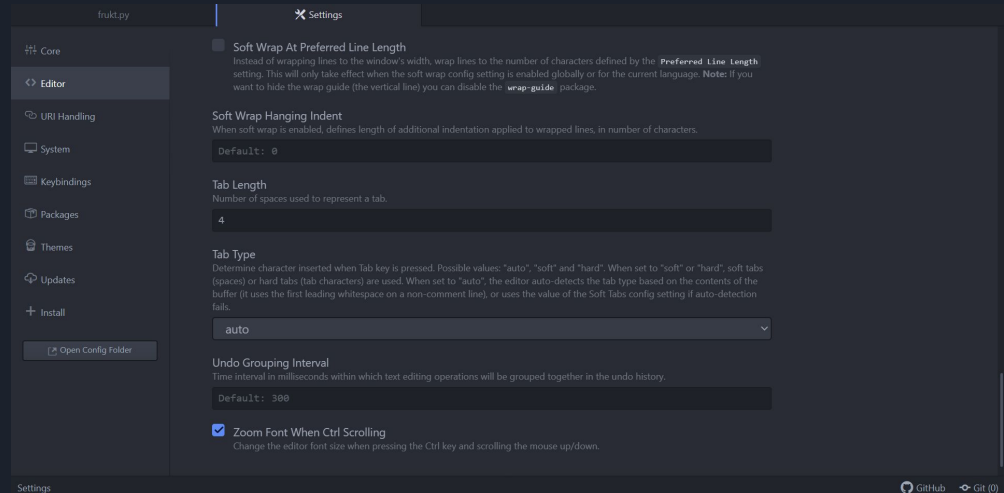
"""

#komentar på en linje

Kodestil

Kodestil og indentering

- Bruk mellomrom i stedet for tab fordi det står i PEP 8 (Style Guide for Python Code) og mellomrom ser likt ut uansett editor/program man bruker
- Om ni vill kan ni lese [PEP 8](#)





Oppgave 0

Diskuter i grupper, forskjeller og likheter i kodeeksemplene under:

<pre>navn = "Nora" alder = 16 if alder >= 18: print(navn, "er myndig.") else: print(navn, "er ikke myndig"</pre>	<pre>NAVN = "Nora" ALDER = 16 if ALDER >= 18: print(NAVN, "er myndig.") else: print(NAVN, "er ikke myndig.")</pre>
<pre>n = "Nora" a = 16 if a >= 18: print(n, "er myndig.") else: print(n, "er ikke myndig.")</pre>	



Data-typer

- Integer (int) = heltall, 3
 - Float (float) = flyttall, 3.14
 - String (str) = tekst, "3"
-
- $5 + 5$ - addisjon
 - "5" + "hei" - konkatenerer tekst
 - $5 + \text{"hei"}$ - feilmelding



Data-typer, sjekke og endre

- `type()`
 - `print(type(5))` -> `<class 'int'>`
 - `print(type(5.4))` -> `<class 'float'>`
 - `print(type("5"))` -> `<class 'str'>`

- `int("3")` eller `int(3.56)` (gir 3, tar bare bort alt bak komma)
- `float("3.14")` eller `float(3)`
- `str(2)`



Evaluering av uttrykk

- Evaluering av uttrykk
 - først høyre sidea av "=", så tilordning til variabel
 - Innifra og ut
 - Ellers skjer evaluering enlig matematiske regler, ganger føre pluss mm.

alder = 6

alder = alder + 3

alder = ((3+2) * 3) + 18



Boolske uttrykk

- Boolske uttrykk - and, or, not. True, False.
 - Finnes kun 2 boolske verdier > sant/True og usant/False.
- Analogi: lysbryter, kan enten være slått av eller på, ingen mellomting og ingen andre verdier mulige..
- If “condition” :
 - $2 < 3$, evaluerer True
 - $3 < 2$, evaluerer False
 - $2 < 4$ and $5 > 2$, evaluerer True
 - $2 > 4$ or $5 > 2$, evaluerer True
 - $\text{not}(2 < 4)$, evaluerer False



Sammenligning

$a == b$	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



Kodeflyt

Vi følger et program linje for linje. Hopper bare over de som vi ikke vil kjøre.

```
lengde = 7
```

```
bredde = 4
```

```
if lengde == bredde:
```

```
    omkrets = 4*lengde
```

```
else:
```

```
    omkrets = (2*lengde) + (2*bredde)
```

```
print("Omkrets: " + str(omkrets))
```



Prosedyrer

Syntaks:

```
def funksjonsNavn() :
```

```
    <det prosedyren skal gjøre>
```

For å kjøre alle kodelinjene i prosedyren ("kalle prosedyren"):

```
funksjonsNavn()
```




Prosedyrer

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

siHallo()

siHallo()

siHallo()



Oppgave 1.

Oppgave 1

Hva evalueres de ulike uttrykkene til?

True and True #

True and False #

True or True #

True or False #

False or False #

not(True) #

not(False) #

Ekstra (litt mer utfordrende):

True and not(True) #

True and not(False) #

not(True) or not(False) #



Oppgave 1.

	Løsning
	True and True # True
	True and False # False
	True or True # True
	True or False # True
	False or False # False
	not(True) # False
	not(False) # True
	<i>Ekstra (litt mer utfordrende):</i>
	True and not(True) # False
	True and not(False) # True
	not(True) or not(False) # True



Oppgave 2

Evaluer følgende uttrykk (True eller False):

$6 > 6$ #

$6 \geq 6$ #

$6 < 6$ #

$6 \leq 6$ #

$6 == 6$ #

$6 != 6$ #



Løsning

Evaluer følgende uttrykk (True eller False):

$6 > 6$ # False


$6 \geq 6$ # True

$6 < 6$ # False

$6 \leq 6$ # True

$6 == 6$ # True

$6 != 6$ # False




Oppgave 3 - individuelt, deretter felles på tavle

Oppgave 3

Finn 3 feil i dette programmet:

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

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



Oppgave 3 - individuelt, deretter felles på tavle

Oppgave 3

Finn 3 feil i dette programmet:

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

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

Løsning

1. Det inni print() skal stå med " " rundt, ellers gir det syntaksfeil.
2. tall < 5 evalueres til true/sant dersom tallet som skrives inn er mindre enn 5 (da printes det ut at tallet er større enn 5). Dette er en logisk feil.
3. Konverterer til int - tillater ikke pannedt enn heltall. Feks. gir man input 5.5 kræsjer programmet, altså vil det gi en runtime error.



Lek

- Gå sammen i gruppe på 3-4 personer
- Skriv 2 enkle prosedyrer (3 min)
- Send arket til neste gruppe
- Skriv et hovedprogram som kaller på hver prosedyre minst en gang. (3min)
- Send arket til neste gruppe
- Numrere i hvilken rekkefølge koden kjører, Og hva som skrives ut i riktig rekkefølge(3min)
- send videre
- Rette forrige gruppes løsning (har dere spørsmål, spørr)