GRUPPETIME UKE 2

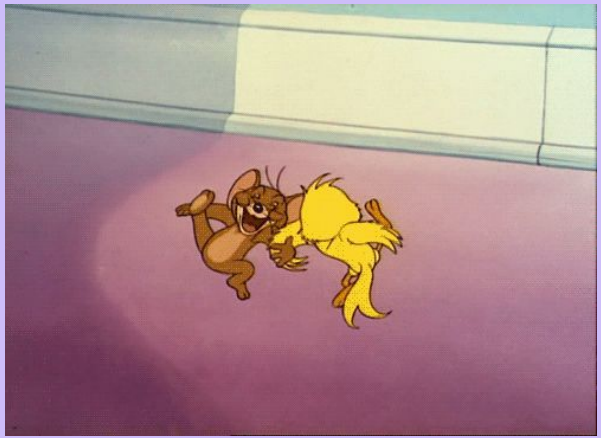
IN1000 GRUPPE 4- 2.02.21

PLAN FOR GRUPPETIMEN

- Litt øvelse fra uke 1
- Jobbe sammen på MetroRetro
- Evaluering av uttrykk
- Kodeflyt
- Datatyper
- Prosedyrer

PROGRAMMING CAN FEEL LIKE...

BUT IT CAN ALSO FEEL LIKE...!



Programmering er kreativt!

Det er som regel mange måter å løse et problem på med programmering.
Etter du har løst et problem, spør deg selv: kunne du ha løst det annerledes?
→ Feks. færre kodelinjer eller annen fremgangsmåte?
→ Diskuter med en venn! Hvordan løste de det?

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

KODESTIL

- Vi bruker PEP8
  - Vær konsistent med navngiving - feks variabel_navn og funksjon_navn()
  - BARE bruk æøå i kommentarer IKKE i variabler, funksjoner, filer etc.
- Spaces > TABS?
  - Mellomrom ser likt ut uansett editor/program man bruker
- Blank lines for å separere relevante kodeblokker
  - Ikke pent/leselig at alt er compressed selv om det er mindre linjer
- Kommenter koden din
  - Oppsummer kort hva programmet skal gjøre øverst i filen
  - Kommenter underveis ovenfor relevant kodeblokk

```
# Slik skrives en kommentar
```

```
mitt_navn = "Nora"
minAlder = 16
if minAlder >= 18:
    print(mitt_navn , "er myndig.")
else:
    print(mitt_navn , "er ikke
myndig")
```

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

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

DISKUTER:

- hva er forskjeller og likheter i kodeeksemplene?

JOBB SAMMEN PÅ METRORETRO

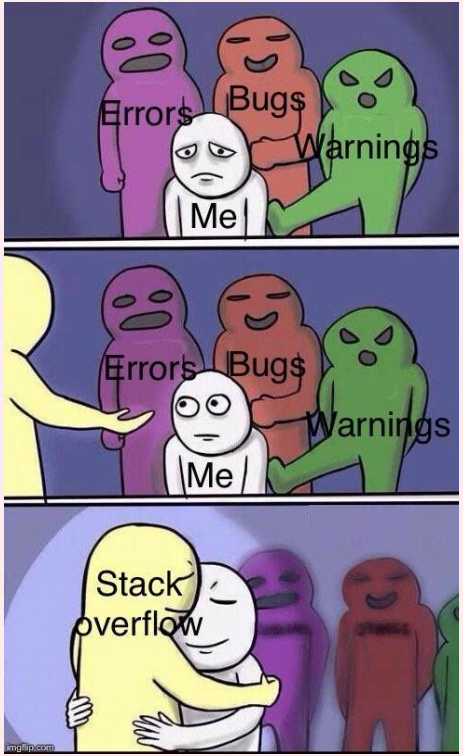
INPUT LAGRES I EN VARIABEL

```
# Programmet venter på bruker input
# men bruker får ingen input beskjed
input()

# Bruker får en beskjed om hva hun bør skrive
# men vi lagrer ikke inputtet
input("Hva heter du?: ")

# Hvorfor bør vi lagre brukerens input?
# ...fordi da kan vi ta i bruk inputet videre!
navn = input("Hva heter du?: ")

if navn == "Harald":
    print("Det heter kongen også!")
elif navn == "Sonja":
    print("Det heter dronningen også!")
else:
    print(f"Hei {navn}!")
```

FEILMELDINGER

- Når noe går galt i et program får vi en feilmelding
- Feilmeldinger består av:
  - Hvor feilen lå (fil og linjenummer)
  - Hvilken type feil det var
  - Akkurat hva som var problemet
- Tre type feil
  - Syntax error
  - Runtime error
  - Logic errors

Prøv å se på Trix, pensum, eller gruppetime slides først!



logiske_feil.py

PASS PÅ LOGISKE FEIL

```
# Hva printes hvis brukeren skriver inn "ja"?
pizza = input("Vil du ha en pizza? ")

if pizza == "ja":
    print("Her har du en pizza!")
if pizza == "nei":
    print("Den er grei.")
else:
    print("Det forstod jeg ikke helt.")
```

elif fikser den logiske feilen

```
>> Her har du en pizza!
>> Det forstod jeg ikke helt.
```

→ *Programmet kjører, men ikke slik vi forventet/ønsket = logisk feil*
→ *Lyst å kode en lignende oppgave? Prøv deg på Trix 02.24!*

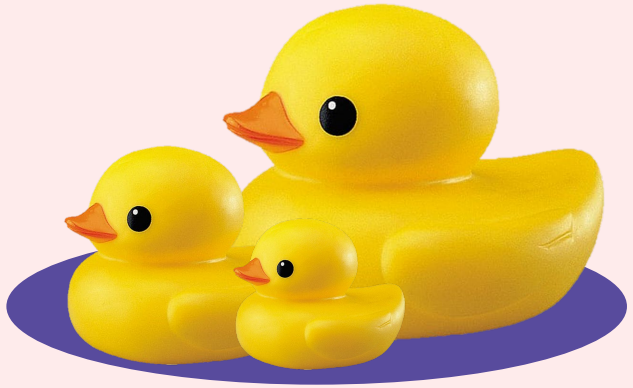
FINN 3 FEIL

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

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

Kan gi ValueError hvis bruker skriver inn en float
Logisk feil
SyntaxError

TIPS

- Les feilmeldingen og identifiser hvilken linje programmet stopper → hvorfor stopper programmet her?
  - Hva har skjedd før denne linjen?
  - Hva var det oppgaven sier at du skal gjøre her?
- Tenk på **typen**! Skulle dette vært en string, int, float, etc.?
- Have you tried explaining it to a rubber duck?
  - Rubber duck debugging
- Tenk på edge cases - hva kan skje i x tilfelle?
- Går det ann å løse problemet på flere måter?
- Ta en pause!



PASS PÅ KODEFLYTEN

en if-sjekk

```
if 5 < 10:
    print("Nr. 1")
elif 5 < 10:
    print("Nr. 2")
>> Nr. 1
```

to if-sjekker

```
if 5 < 10:
    print("Nr. 1")
if 5 < 10:
    print("Nr. 2")
>> Nr. 1
>> Nr. 2
```

KODEFLYT: TRIX 2.11

```
    1 # Vis kodeflyten hvis brukeren taster inn 50
1|  2 pris = 50
2|  3 tekst = input("Skriv inn alder: ")
3|  4 alder = int(tekst)
    5
4|  6 if alder < 12 or alder > 67:
    7     print("Du må betale", pris*0.5, "kr")
5|  8 else:
6|  9     print("Du må betale", pris, "kr")
   10
7|11 print("Ha en fin dag!")
```

→ *For å se kodeflyten kan du kjøre koden i PythonTutor!*

TRUE/FALSE?

```
1.  True and True
2.  True and False

3.  True or True
4.  True or False
5.  False or False

6.  not(True)
7.  not(False)

#Ekstra (utfordring)
8.  True and not(True)
9.  True and not(False)
10. not(True) or not(False)
```

→ Et boolsk uttrykk evaluerer til enten True eller False

AND

True	True	**True**
True	False	**False**
False	True	**False**
False	False	**False**

OR

True	True	**True**
True	False	**True**
False	True	**True**
False	False	**False**

NOT

True	**False**
False	**True**

→ Vi kan sette sammen boolske uttrykk med **and** og **or**

SAMMENLIGNINGSOPERATØRER

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

TYPEKONVERTERING

```
str(4)              # konvertere til string > “4”
int("5")            # konvertere string til int > 5
int("   6   ")      # konvertere string til int, mellomrom vil bli ignorert
float("17.25")# konvertere string til float > 17.25
float("3x6")        # gir Error! Siden x ikke er et tall.
```

Usikker på typen? Da kan du sjekke med type()

```
navn = "Siri"
print(type(navn))      >> <class 'str'>
```

→ Lyst å øve på en lignende oppgave? Prøv deg på Trix 02.02!

PROSEDYRER uten returverdi eller parametere

- Prosedyrer uten returverdi eller parametere er nyttig dersom man skal gjøre det samme flere ganger → unngår å repetere samme kode
  - def *prosedyre_navn() :*

    <det prosedyren skal gjore>
- NB! Prosedyren må være definert før man kaller på den

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

si_hallo()
```

prosedyre.py

PROSEDYRE EKSEMPEL

```
    1 # What will print if the user inputs "Anna"?
  1|2 def king_greeting():
    3     print("O'king, welcome!")
    4
  2|5 def queen_greeting():
    6     print("Majesty queen, welcome!")
    7
3,9|8 def commoner_greeting():
 10|9     print("Welcome, peasant.")
   10
  4|11 name = input("What is your name?\n>> ")
   12
  5|13 if name == "Harald":
   14     king_greeting()
  6|15 elif name == "Sonja":
   16     queen_greeting()
  7|17 else:
  8|18     commoner_greeting()
```

→ Lyst å kode en lignende oppgave? Prøv deg på Trix 02.22 eller Trix 2.17!

FINN FEILEN

```
# What will print if the user inputs "Anna"?
name = input("What is your name?\n>> ")

if name == "Harald":
    king_greeting()
elif name == "Sonja":
    queen_greeting()
else:
    commoner_greeting()

def king_greeting():
    print("O'king, welcome!")

def queen_greeting():
    print("Majesty queen, welcome!")

def commoner_greeting():
    print("Welcome, peasant.")
```

Prosedyrer må
være definert
før vi kaller på de!

```
Traceback (most recent call last):
  File "prosedyre.py", line 8, in <module>
    commoner_greeting()
NameError: name 'commoner_greeting' is not
defined
```

PROSEDYRER og veien videre

- Prosedyre - uten parametre og returverdi
- Prosedyre - med parametre
- Funksjon - med returverdi
- metoder - handlinger vi kan gjøre med objekter (OO)

PROSEDYRE ØVELSE I METRORETRO

1. Skriv to enkle prosedyrer som dere kan kalle hva som helst
2. Bytt ark i MetroRetro med neste gruppe til høyre

   1 → 2 2 → 3 3 → 4 4 → 1

   a. Skriv et hovedprogram hvor dere kan legge til hva dere vil, men dere må ha minst ett kall på hver av prosedyrene skrevet på arket
3. Bytt ark i MetroRetro med neste gruppe til høyre
   a. Nummerer kodeflyten (i hvilken rekkefølge skjer ting når programmet kjøres) og skriv utskriften i riktig rekkefølge
4. Bytt ark i MetroRetro med neste gruppe til høyre
   a. Rett forrige gruppe sin løsning hvis det er noe å rette

Vil du strekke deg til flere utfordringer?

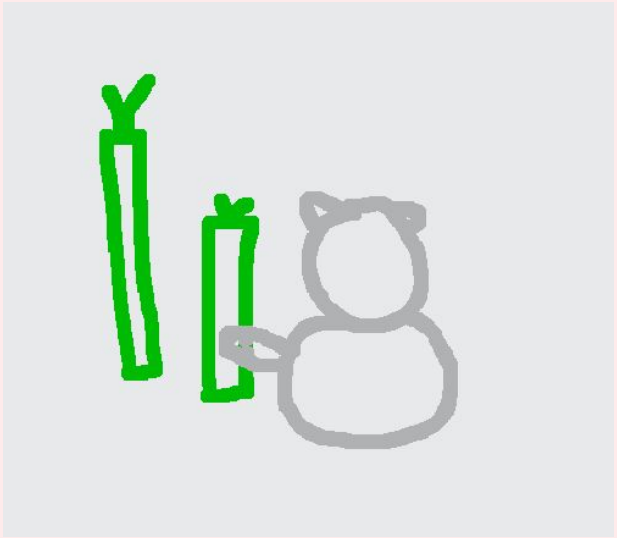
https://www.codewars.com/
https://projecteuler.net/
https://open.kattis.com/

KONTAKT



...

sirisoll@uio.no
sirisoll på Mattermost



UKENS MESTERVERK: Panda med bambus