

# Øvingsforelesning i Python (TDT4110)

Tema: Øving 2, Betingelser, if/elif/else

Kristoffer Hagen

# Oversikt

- Praktisk informasjon
- Gjennomgang av Øving 1
- Oppgaver for Øving 2

# Praktisk

- Bruke andre studasser (oppklaring)
- Flere som bruker Python 2
- `print('Vekt paa bilen [kg]:', 750)`
  - `('Vekt paa bilen [kg]: ', 750)` -Python 2
  - `Vekt paa bilen [kg]: 750` -Python 3

# Oversikt

- Praktisk informasjon
- Gjennomgang av Øving 1
- Oppgaver for Øving 2

# Gjennomgang øving 1

- Meldinger
- Scoping

# Meldinger

1. Definer `logg()` funksjonen med de tre parameterene
2. Skriv `print()` funksjonen som `logg()` skal kjøre
3. Definer `main()` funksjonen
4. Inne i `main`, skriv de 6 meldingene vha `logg()`
5. Kall `main()` funksjonen

# Meldinger del 2

1. Opprett en (global) variabel kalt meldingNr
2. Si til logg() at meldingNr er en global variabel og at den skal økes med en hver gang logg() brukes (var += 1)
3. Endre på print() i logg() slik at nummeret skrives ut

# Scoping

- Globale variabler kan alltid leses
- Lokale variabler med samme navn brukes over globale
- Når en variabel blir sendt som parameter til en funksjon er det verdien av variabelen som sendes



# Oversikt

- Praktisk informasjon
- Om øvingsforelesninger
- Oppgaver for Øving 2

# PyCharm Editor

- <https://www.jetbrains.com/pycharm/>
- Mer stabilt på MAC enn IDLE
- Mer oversiktelig når en blir kjent til det

# Setup av PyCharm

- Lag et nytt prosjekt og gi det et navn, f.ek «øvingsforelesning1»
- Lag så en ny python fil og begynn å programmer
- Skriv et program som printer «æ ø å Æ Ø Å»

# Setup av PyCharm

- Dersom programmet krasjer må du endre noe
- Gå til file->settings->editor->file encodings
  - Sett IDE og PROJECT ENCODING til UTF-8
- Prøv å kjør programmet igjen
- Dersom du får feil igjen, start programmet med (øverst)  
`# -*- coding: utf-8 -*-`
- Dersom det fortsatt ikke fungerer må du unngå norske bokstaver

# Oppgave 1

- Lag et program som tar inn et tall via input, så printer om tallet er større eller mindre enn 10
- (husk int/float før input)
- Syntaks:           if (logisk uttrykk):  
                          kode her  
                  else:  
                          kode her

# Oppgave 1.1

- Gjør slik at programmet skriver ut dersom tallet er lik 10 også
  
- Hint: elif (logisk uttrykk):

# Oppgave 1.2

- Lag et program som tar inn to tall via input (vi kaller dem tall1 og tall2), og skriver ut dersom tallene er like

# Oppgave 1.3

- Dersom tallene er ulike, print dette også



# Oppgave 1.4

- Dersom tallene er ulike, print om det andre tallet (tall2) er høyere eller lavere enn tall1
- 'Tall 2 er for lavt' eller 'Tall 2 er for høyt'

# Oppgave 1.5 **\*\*Vanskelig\*\***

- Dersom tall 2 er for høyt eller lavt, skriv ut om det er veldig mye større, eller mindre (differanse over 100)

# Oppgave 1.6

- Gjør at tall1 blir satt til et tilfeldig tall mellom 1 og 1000
- Import random
- `A = random.randint(min,max)`

# Oppgave 1.7

- (Liten tyvstart for å få laget dette til et spill)
- Gjør at programmet repeteres til det rette tallet er funnet
- Hint: while True:
- Bruk break når tallet er funnet

# Frivillige oppgaver for senere

- Gjør at programmet sier ifra viss differansen er veldig liten (under 10 f.ek)
- Lag en variabel som teller antall forsøk du bruker

# Oppgave 2

- Lag et program som endrer et heltall (fra 1 til 5) til en streng og printer strengen
- 1 blir til 'en', 4 blir til 'fire'

# Oppgave 2.1

- Gjør programmet om til en funksjon som tar tallet inn som en parameter
- Hint: `def int_til_streng(tallet):`

# Oppgave 2.2

- Lag og kall en main() metode som skriver ut:  
En  
To  
Tre  
Fire  
Fem



## Oppgave 2.3 **\*\*Vanskelig\*\***

- Lag en ny metode, `int_til_streng_utvidet()`, som tar inn et tall på formen `x0x` der `x` er mellom 1 og 5, og printer ut tallet som tekst
- Eksempel: `505 = FemhundreogFem`
- Hint: `503//100 = 5`, `503%100 = 3`
- Gjenbruk forrige funksjon (`int_til_streng`)

# Oppgave 2.3 (litt hjelp)

- Del opp i fire deler:
  1. Dersom tallet er under hundre, bruk gamle funksjonen
  2. Dersom tallet er over hundre, bruk `tallet//100` for å fine «hvor mange hundre», så bruk gamle funksjonen for å printe dette
  3. Print «hundreog»
  4. Finn siste sifferet med `tallet%100` (rest etter det deles på hundre), så bruk gamle funksjonen for å printe dette

To  
Hundreog  
Fem

# Spørsmål?

- Send meg evt. spørsmål og tilbakemelding (relatert til øvingsforelesningene) på [kristoffer.hagen@idi.ntnu.no](mailto:kristoffer.hagen@idi.ntnu.no)