

Velkommen til gruppetime i IN1000



28. august 2020
Jessie Yue Guan

Planen for i dag

- Litt repetisjon
- Logiske operasjoner
- Sjekke variabeltyper
- Konvertere variabeltyper
- Prosedyrer
- Feilmeldinger
- Kodeflyt
- Sammenligning av datoer
- Kahoot! :D

String

- Forkortes ofte som "str"
 - Tekstverdier
 - Kan være bokstaver, tall, eller spesialtegn
 - Husk å bruke ' ' eller " " eller "" ""
- For eksempel:
 - **bokstav = "k"**
 - **tegn = "!"**
 - **setning = "Hei " + brukernavn**
 - **sporsmaal = "Er du 18 år gammel?"**
 - **tekst = "sWo32)"#"**

Hva om jeg vil ha " " i strengen?

1. `print("En setning som trenger anførelstegn")`
 2. `print("""En setning som trenger anførelstegn""")`
 3. `print("\En setning som trenger anførelstegn\"")`
- Alle alternativer fører til samme output:
 - `"En setning som trenger anførelstegn"`

Kombinasjoner av if-setninger

- `a = 5`
- `if (a > 0):`
- `print("Sann")`
- `if (a < 0):`
- `print("Usann")`

- `if (a > 1):`
- `print("Sann")`
- `else:`
- `print("Usann")`
- `if (a < 1):`
- `print("Usann")`
- `else:`
- `print("Sann")`

Kombinasjoner av if-setninger

- `b = 25`

- `if b > 20:`

- `print("Sann")`

- `elif b < 20:`

- `print("Usann")`

- `if b < 21:`

- `print("Usann")`

- `elif b > 21:`

- `print("Sann")`

- `if b < 22:`

- `print("Usann")`

- `elif b < 23:`

- `print("Usann")`

- `if b > 22:`

- `print("Sann")`

- `elif b > 23:`

- `print("Sann")`

Kombinasjoner av if-setninger

- `c = 35`
- `if c > 30:`
- `print("Sann")`
- `elif c < 30:`
- `print("Usann")`
- `else:`
- `print("Usann")`

- `if c < 31:`
- `print("Usann")`
- `elif c > 31:`
- `print("Sann")`
- `else:`
- `print("Usann")`

- `if c > 32:`
- `print("Sann")`
- `elif c > 33:`
- `print("Sann")`
- `else:`
- `print("Usann")`

- `if c < 32:`
- `print("Usann")`
- `elif c < 33:`
- `print("Usann")`
- `else:`
- `print("Sann")`

Oppgaver med relasjonelle operasjoner

$0 == 0$

True

$0 == 1$

False

$1 != 0$

True

$1 != 1$

False

$2 < 3$

True

$3 < 2$

False

$5 > 4$

True

$4 > 5$

False

$6 \leq 7$

True

$7 \leq 6$

False

$9 \geq 8$

True

$8 \geq 9$

False

Kodestil

- Variabelnavn burde si noe om hva verdien er
- Vær organiserte og detaljorienterte
- Gi tydelige instruksjoner til brukeren
- Gi logiske tilbakemeldinger til brukeren

Plusstegnet og kommategnet

- Hva er forskjellen mellom + og , når vi printer?
- Pluss er veldig streng og gir ikke mellomrom
- Komma er veldig fleksibel og gir mellomrom

- Uten variabler
 - `print(2 + 3)`
 - `print("2" + "3")`

 - `print(2, 3)`
 - `print("2", "3")`

- Med variabler:
 - `a = 4`
 - `b = 6`

 - `print(a + b)`
 - `print("a" + "b")`

 - `print(a, b)`
 - `print("a", "b")`

Spørsmål?

- Ikke vær redd for å spørre, det finnes ingen dumme spørsmål! 😊

Logiske operasjoner

- Brukes på booleans
- Resulterer i en boolean
- Består av følgende:
 - negasjon (**not**) som betyr "ikke"
 - konjunksjon (**and**) som betyr "både og"
 - disjunksjon (**or**) som betyr "enten eller"

Logiske operasjoner - not

- La oss si at vi har en boolean variabel som heter **min_bool**
- Hvis vi setter **True** som verdien til **min_bool** vil **not min_bool** bli **False**
- Hvis vi setter **False** som verdien til **min_bool** vil **not min_bool** bli **True**

min_bool	not min_bool
True	False
False	True

Logiske operasjoner - and

- La oss si at vi har en boolean variabel kalt **x** og en boolean variabel kalt **y**
- Hvis *både* **x** og **y** har verdien **True**, så vil verdien av **x and y** også bli **True**
- I alle andre tilfeller vil **x and y** bli **False**

x	y	x and y
True	True	True
False	True	False
True	False	False
False	False	False

Logiske operasjoner - or

- La oss si at vi har en boolean variabel kalt **x** og en boolean variabel kalt **y**
- Hvis *både* **x** og **y** har verdien **False**, så vil verdien av **x or y** også bli **False**
- I alle andre tilfeller vil **x or y** bli **True**

x	y	x or y
True	True	True
False	True	True
True	False	True
False	False	False

Logiske operasjoner - rekkefølge

- Det er alltid **not** som blir evaluert først
 - For eksempel: **not True or True** blir altså **False**
- Deretter er det **and** som blir evaluert
 - For eksempel: **True and False or True** blir altså **False**
- Til slutt er det **or** som blir evaluert
 - For eksempel: **not False or False** blir altså **True**

Oppgaver med logiske operasjoner

not False

not True

True and True

True and False

False and True

False and False

True or True

True or False

False or True

False or False

True

False

True

False

False

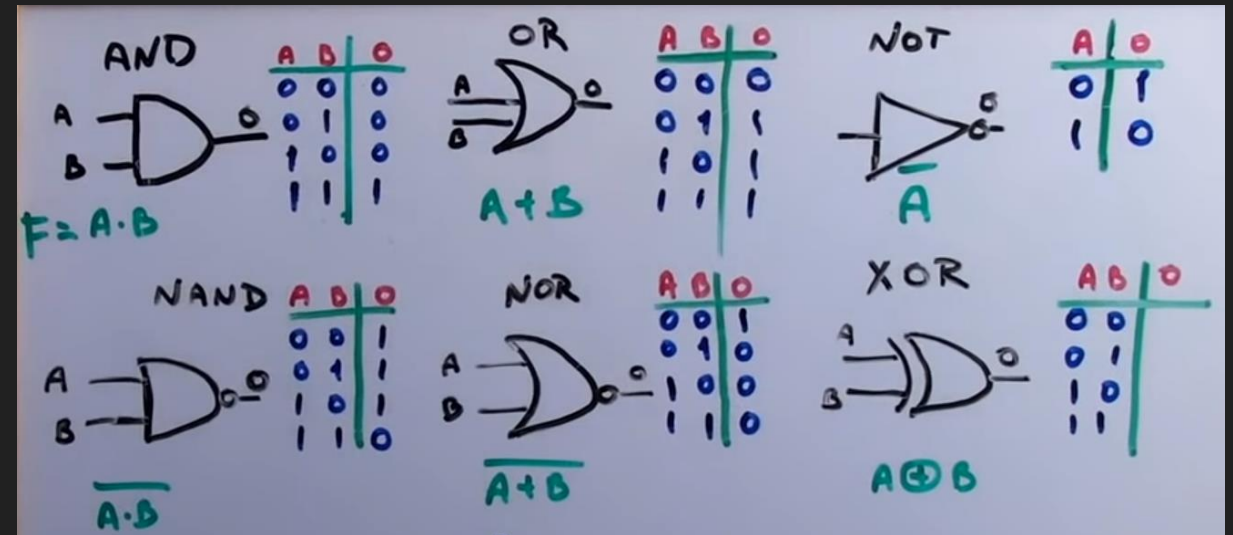
False

True

True

True

False



Spørsmål?

- Ikke vær redd for å spørre, det finnes ingen dumme spørsmål! 😊

Hvordan sjekke variabeltyper?

- `type(True)`
- `type(2)`
- `type(3.14)`
- `type("Hei")`
- `type('Jessie')`
- `type("""1\n2\n3""")`
- `#bool`
- `#int`
- `#float`
- `#str`
- `#str`
- `#str`
- Du kan bruke `type()` med for eksempel `print()` eller `if`-setninger

Hvordan konvertere til andre variabeltyper?

○ Konvertering fra str til andre variabeltyper

- `tekst1 = "100"`
- `tekst2 = "99.9"`
- `tekst3 = "True"`

- `heltall = int(tekst1)`
- `print(type(heltall))`

- `desimaltall = float(tekst2)`
- `print(type(desimaltall))`

- `boolskverdi = eval(tekst3)`
- `print(type(boolskverdi))`

○ Konvertering mellom int og float

- `tall1 = 10`
- `tall2 = 9.9`

- `tall3 = float(tall1)`
- `print(tall3)`
- Vi får 10.0

- `tall4 = int(tall2)`
- `print(tall4)`
- NB! Vi får 9 og ikke 10!

Hvordan konvertere input til andre variabeltyper?

- Husk at input ALLTID blir tolket som STRING uansett hva du skriver
 - `alder = int(input("Hvor gammel er du? "))`
 - `print("Du svarte", alder, "\n")`

 - `hoyde = float(input("Hvor hoy er du? "))`
 - `print("Du svarte", hoyde, "\n")`

 - `student = eval(input("Er du student? Svar True eller False: "))`
 - `print("Du svarte", student, "\n")`

Prosedyrer

- `def min_prosedyre():` ← Definere prosedyren
 - `print("Hei")` ← Fulle opp prosedyren
- `min_prosedyre()` ← Kjøre prosedyren

Hvordan lese feilmeldinger?

- **navn = Geir**

- NameError: name 'Geir' is not defined

- **if 2>1:**

- **print("ja")**

- IndentationError: expected an indented block

- **print("ja"**

- SyntaxError: invalid syntax

- **alder = int(input("Hvor gammel er du?"))**

- ValueError: invalid literal for int() with base 10: "

Oppgave med feilmeldinger

- `tall = int(input("Skriv inn et tall: "))`
- `if tall < 5:`
- `print(Tallet er større enn 5)`
- `else:`
- `print(Tallet er ikke større enn 5)`

Kodeflyt

- `def kvadrer() :`
- `tall = float(input("Skriv inn et tall: "))`
- `print("Produkt: ", tall*tall)`

- `def velkommen() :`
- `navn = input("Hva heter du? ")`
- `print("Velkommen til IN1000", navn)`

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

- `siHallo()`
- `velkommen()`
- `print("Klar for å kvadrere litt?")`
- `kvadrer()`

- `> Hallo!`
- `> Hva heter du? Kristine`
- `> Velkommen til IN1000 Kristine`
- `> Klar for å kvadrere litt?`
- `> Skriv inn et tall: 2`
- `> Produkt: 4.0`

Kodeflyt

- `def kvadrer() :`
- `tall = float(input("Skriv inn et tall: "))`
- `print("Produkt: ", tall*tall)`

- `def velkommen() :`
- `navn = input("Hva heter du? ")`
- `print("Velkommen til IN1000", navn)`

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

- `siHallo()`
- `velkommen()`
- `print("Klar for å kvadrere litt?")`
- `kvadrer()`

- `> Hallo!`
- `> Hva heter du? Kristine`
- `> Velkommen til IN1000 Kristine`
- `> Klar for å kvadrere litt?`
- `> Skriv inn et tall: 2`
- `> Produkt: 4.0`

Hvordan gjøre om input til andre variabeltyper med sikkerhet?

- `navn = input("Hva heter du? ")`
- `print("Hei", navn)`

- `def sporsmaal():`
- `svar = input("Hvor hoy er du i cm? ")`
- `try:`
- `hoyde = float(svar)`
- `print("Da er du", hoyde/2.54, "inches hoy")`
- `except ValueError:`
- `print("Dette er ikke et tall!")`
- `sporsmaal()`

- `sporsmaal()`

Spørsmål?

- Ikke vær redd for å spørre, det finnes ingen dumme spørsmål! 😊