Problemløsing med typer og prosedyrer

Frøya Lydersen

Uke 2, time 2.

Gruppe 6

Noen ting vi ikke fikk tid til forrige time!

Literaler

```
1  x = "frosker" + " er " + "kule"
2  y = 10 + 19
3  z = True
4  w = (("Ost" + "epop") == "katt") and (2+1)>2
```

Literaler

```
1  x = "frosker" + " er " + "kule"
2  y = 10 + 19
3  z = True
4  w = (("Ost" + "epop") == "katt") and (2+1)>2
```

Kodeflyt

Rett og slett et generelt begrep for hvordan koden “flyter”.

- Hvordan uttrykk evalueres
- Hvordan python hopper over deler av en if-statement (programsetning)
- Hvordan python hopper inn og ut av prosedyrer
- Osv.

Logiske/boolske operatorer

or

```
False or False = False
False or True  = True
True  or False = True
True  or True  = True
```

and

```
False and False = False
False and True  = False
True  and False = False
True  and True  = True
```

not

```
not False = True
not True  = False
```

Eksempel: not

```
1  har_lekser = False
2  if not har_lekser:
3      print("Se på serie!")
```

Eksempel: not

```
1  har_lekser = False
2  if not har_lekser:
3      print("Se på serie!")
```

```
1  har_billett = True
2  har_legitimasjon = False
3  if not (har_billett and har_legitimasjon):
4      print("Kommer ikke inn!")
```

Sammenlignings-operatorer

- ==
- !=
- >
- <
- >=
- <=

Sammenlignings-operatorer

```
1  pin = input("Skriv PIN:")
2  if pin != "0316":
3      print("Inntrenger!!")
4  else:
5      print("Velkommen :)")
```

- ==
- !=
- >
- <
- >=
- <=

Sammenlignings-operatorer

```
1  pin = input("Skriv PIN:")
2  if pin != "0316":
3      print("Inntrenger!!")
4  else:
5      print("Velkommen :)")
```

- ==
- !=
- >
- <
- >=
- <=

```
1  bil_vekt = 1.36 # tonn
2  if bil_vekt >= 0.6 and bil_vekt <= 2.2:
3      print("Innenfor normalvekt for jekken!")
```



Mer naturlig kode

Pythonkoden vår kan lett bli vanskelig å lese. Da kan det hjelpe å flytte ut deler av et uttrykk til variabler.

Mer naturlig kode

Pythonkoden vår kan lett bli vanskelig å lese. Da kan det hjelpe å flytte ut deler av et uttrykk til variabler.

```
1 if int(input("Hvor gammel er du? ")) >= 18 and input("Har du
  ID? y/n: ") == "y":
2     print("Du kommer inn!")
```

Mer naturlig kode

Pythonkoden vår kan lett bli vanskelig å lese. Da kan det hjelpe å flytte ut deler av et uttrykk til variabler.

```
1  if int(input("Hvor gammel er du? ")) >= 18 and input("Har du
   ID? y/n: ") == "y":
2      print("Du kommer inn!")
```

Mer naturlig kode

Pythonkoden vår kan lett bli vanskelig å lese. Da kan det hjelpe å flytte ut deler av et uttrykk til variabler.

```
1  alder = int(input("Hvor gammel er du? "))
2  if alder >= 18 and input("Har du ID? y/n: ") == "y":
3      print("Du kommer inn!")
```

Mer naturlig kode

Pythonkoden vår kan lett bli vanskelig å lese. Da kan det hjelpe å flytte ut deler av et uttrykk til variabler.

```
1  alder = int(input("Hvor gammel er du? "))
2  if alder >= 18 and input("Har du ID? y/n: ") == "y":
3      print("Du kommer inn!")
```

Mer naturlig kode

Pythonkoden vår kan lett bli vanskelig å lese. Da kan det hjelpe å flytte ut deler av et uttrykk til variabler.

```
1  alder = int(input("Hvor gammel er du? "))
2  myndig = alder >= 18
3
4  if myndig and input("Har du ID? y/n: ") == "y":
5      print("Du kommer inn!")
```

Mer naturlig kode

Pythonkoden vår kan lett bli vanskelig å lese. Da kan det hjelpe å flytte ut deler av et uttrykk til variabler.

```
1  alder = int(input("Hvor gammel er du? "))
2  myndig = alder >= 18
3
4  if myndig and input("Har du ID? y/n: ") == "y":
5      print("Du kommer inn!")
```

Mer naturlig kode

Pythonkoden vår kan lett bli vanskelig å lese. Da kan det hjelpe å flytte ut deler av et uttrykk til variabler.

```
1    alder = int(input("Hvor gammel er du? "))
2    myndig = alder >= 18
3    har_id = input("Har du ID? y/n: ") == "y"
4
5    if myndig and har_id:
6        print("Du kommer inn!")
```

Mer naturlig kode

Pythonkoden vår kan lett bli vanskelig å lese. Da kan det hjelpe å flytte ut deler av et uttrykk til variabler.

```
1    alder = int(input("Hvor gammel er du? "))
2    myndig = alder >= 18
3    har_id = input("Har du ID? y/n: ") == "y"
4
5    if myndig and har_id:
6        print("Du kommer inn!")
```

Mer naturlig kode

Pythonkoden vår kan lett bli vanskelig å lese. Da kan det hjelpe å flytte ut deler av et uttrykk til variabler.

```
1 if int(input("Hvor gammel er du? ")) >= 18 and input("Har du
  ID? y/n: ") == "y":
2     print("Du kommer inn!")
```

Feilmeldinger

Det finnes hovedsakelig 3 typer feil:

1. Syntaks-feil - *små skrivefeil, kræsj før den kjører*
2. Runtime-feil - *kræsj mens den kjører*
3. Logiske feil - *kjører uten feil, gjør ikke det du vil!*

De første to typene feil gir eksplisitte feilmeldinger som:

- `NameError`
- `SyntaxError`
- `IndentationError`
- `ZeroDivisionError`

Syntaksfeil

Manglende kolon:

```
1  alder = 16
2  if alder < 18
3      print("Kommer ikke inn!")
```

```
  File "/home/snuppy/Nextcloud/h26/in1000ta/uke2/t2/run/syntaksfeil1.py", line 2
    if alder < 18
                ^
SyntaxError: expected ':'
```

Syntaksfeil

Feil innrykk:

```
1  if True:
2  print("hei!")
```

```
  File "/home/snuppy/Nextcloud/h26/in1000ta/uke2/t2/run/syntaksfeil2.py", line 2
    print("hei!")
    ^^^^^
IndentationError: expected an indented block after 'if' statement on line 1
```

Syntaksfeil

Ubalansert parantes:

```
1  print(7+(5/2)-10))
```

```
  File "/home/snuppy/Nextcloud/h26/in1000ta/uke2/t2/run/syntaksfeil3.py", line 1
    print(7+(5/2)-10))
                     ^
SyntaxError: unmatched ')'
```

Syntaksfeil

Feilstaving:

```
1  if False:
2      print("hei")
3  eleif True:
4      print("nei")
```

```
  File "/home/snuppy/Nextcloud/h26/in1000ta/uke2/t2/run/syntaksfeil4.py", line 3
    eleif True:
    ^^^^^
SyntaxError: invalid syntax. Did you mean 'elif'?
```

Runtime-feil

Typefeil:

```
1  alder = "20"
2  print("alder" + 5)
```

```
Traceback (most recent call last):
  File "/home/snuppy/Nextcloud/h26/in1000ta/uke2/t2/run/runtimefeil1.py", line 2, in <module>
    print("alder" + 5)
          ~~~~~~~~^~~
TypeError: can only concatenate str (not "int") to str
```

Runtime-feil

Deling på null:

```
1  poeng = 10
2  antall = 0
3  print(poeng / antall)
```

```
Traceback (most recent call last):
  File "/home/snuppy/Nextcloud/h26/in1000ta/uke2/t2/run/runtimefeil2.py", line 3, in <module>
    print(poeng / antall)
          ~~~~~~^~~~~~~~
ZeroDivisionError: division by zero
```

Runtime-feil

Bruk av variabel som ikke er tilordnet noe:

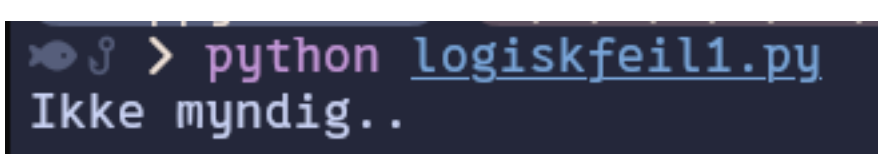
```
1  print(gweegh + 50)
```

```
Traceback (most recent call last):
  File "/home/snuppy/Nextcloud/h26/in1000ta/uke2/t2/example_problem_code/syntaksfeil4.py", line 1, in <module>
    print(gweegh + 50)
          ^^^^^^
NameError: name 'gweegh' is not defined
```

Logiske feil

Off-by-one:

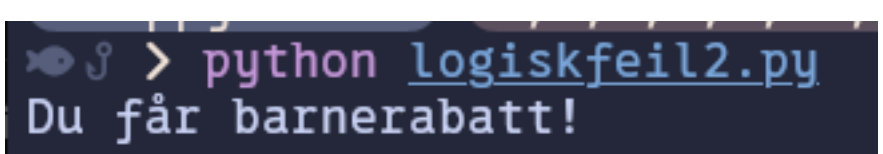
```
1  alder = 18
2  if alder > 18:
3      print("Myndig!")
4  else:
5      print("Ikke myndig..")
```



Logiske feil

Feil i vår logikk:

```
1  barnerabatt_brukt = False
2  alder = 16
3
4  if alder < 15 or not barnerabatt_brukt:
5      print("Du får barnerabatt!")
```



`assert`

- Til og med erfarne programmerere skriver kode som har logiske feil.
- Vi vil mye heller at koden vår skal kræsje og gi beskjed om at noe er galt, enn at den kjører videre og gir feil output!
- `assert` lar oss sjekke noe, og kræsje der og da hvis det ikke er tilfelle.

Vi tar et eksempel!

Funksjoner som evaluerer til noe

- Noen av funksjonene vi har snakket om, `int()`, `str()`, osv, `input()`, evaluerer til en verdi.
- Merk at `print()` ikke evaluerer til en verdi.

Vi tar en quiz på menti!

Vi løser en oppgave sammen!

–Pause!–

Vi tar en alias :D

Vi tar en alias :D



Vi gjør litt parprogrammering!

–Exit ticket–