

IN1000 – seminarartime uke 2

Gruppe 14 – Seminarrom Logo

Opprop og div. info

- Er du syk eller har luftveissymptomer? Hold deg hjemme! Finnes en haug med digitale tilbud på IFI
- Kontaktinfo: ahmedws@uio.no, eller huk tak i meg utenom timene
- Bruk tilbakemeldingene fra oblig 1 i videre arbeid

Agenda

- God kodestil, variabelnavn, tab eller mellomrom, [PEP-8](#)
- Feilmeldinger, feil som ikke gir feilmeldinger (logiske feil)
- Boolske uttrykk
- Typekonvertering
- Prosedyrer uten parametere og returverdi
- En morsom øvelse

Diskuter forskjeller og likheter i kodesnuttene

<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>	

God kodelstil (kodeskikk)

- Små forbokstaver på variabelnavn
- Tydelige variabelnavn, selvforklarende navn kan være til stor nytte
- CamelCase vs. _underscores, hva skal man bruke?
 - For de interesserte: [klikk her](#)
- [PEP-8](#)

Feilmeldinger

- To hovedtyper:
 - Syntaksfeil
 - Unntak/exceptions, mer om dette senere i kurset
- Feil som ikke gir tilbakemeldinger (logiske feil)
- Livekoding av et eksempel
- Oblig 1?

Syntaksteil:

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

```
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'
```

Boolske uttrykk

- And
- Or
- Not
- True, False – kan se på det som en lysbryter, enten av eller på
- And: begge må være sanne, or: holder med en sann verdi, not: snur om

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) #

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) #

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

Ekstra (litt mer utfordrende):

True and not(True) # False

True and not(False) # True

not(True) or not(False) # True

Oppgave 2

Evaluer følgende uttrykk (True eller False):

$6 > 6$ #

$6 \geq 6$ #

$6 < 6$ #

$6 \leq 6$ #

$6 == 6$ #

$6 != 6$ #

Typekonvertering

- `str()` – konverterer til string
- `int()` – konverterer string til heltall
- `float()` – konverterer string til flyttall
- NB: kan gi feilmeldinger
- Livekoding

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

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 størret enn 5). Dette er en logisk feil.
3. Konverterer til int - tillater ikke annet enn heltall. Feks. gir man input 5.5 kræsjer programmet, altså vil det gi en runtime error.

Prosedyrer uten parametere

- Et verktøy for å unngå repetisjon av kode
- Hva er en parameter?
- Repetisjon av kodeflyt

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

```
siHallo()
```

```
siHallo()
```

```
siHallo()
```

Spørsmål?

Prosedyre-lek