Tredje gruppetime!!

Gjennomgang av ukens pensum og felles oppgaveløsning :)

Triera Gashi - trierag@ifi.uio.no



Repetisjon

Repetisjon: input

Syntaks:
navn = input()

Dersom man vil gi en utskrift før input:
navn = input(“Hva heter du?”)

Det som skrives som input (frem til enter trykkes) lagres i variabelen navn.

For å skrive ut på ny linje kan man bruke "\n" for newline.

Slik kan du få brukeren til å skrive inputet på en ny linje:
navn = input(“Hva heter du?\n”)

2

```
navn = “Nora”
alder = 16
if alder >= 18:
        print(navn, “er myndig.”)
else:
      print(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.”)
```

```
navn=“Nora”
alder=16
if alder>=18:
      print(navn,“er myndig.”)
else:
      print(navn,“er ikke myndig”)
```

Hva er forskjellene på kodesnuttene?

Type-konvertering

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

Oblig 1 - Oppgave 3

Boolske uttrykk

Boolske verdier

- Finnes kun to boolske verdier: True og False
- Boolske operatorer: And, or og not
- Sammenligningsoperatorer: <, >, ==, <=, >=
- Boolske operatorer og sammenligningsoperatorer lager boolske uttrykk
- Boolske uttrykk evaluerer til en boolsk verdi

Boolske operatorer

- For at **and** skal gi sant må begge uttrykk være sanne
- For at **or** skal gi sant må *minst* ett av uttrykkene være sanne
- **not** evaluerer alltid til det motsatte

Boolean Operators

AND

A	B	A AND B
True	True	True
True	False	False
False	True	False
False	False	False

OR

A	B	A OR B
True	True	True
True	False	True
False	True	True
False	False	False

NOT

A	NOT A
True	False
False	True

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)      #
```

Ekstra ekstra utfordring:
Hva evaluerer følgende uttrykk til:

(not False or True and not False)

(not False or True and not False)

er det samme som

(not False) or (True and (not False))

not > and > or

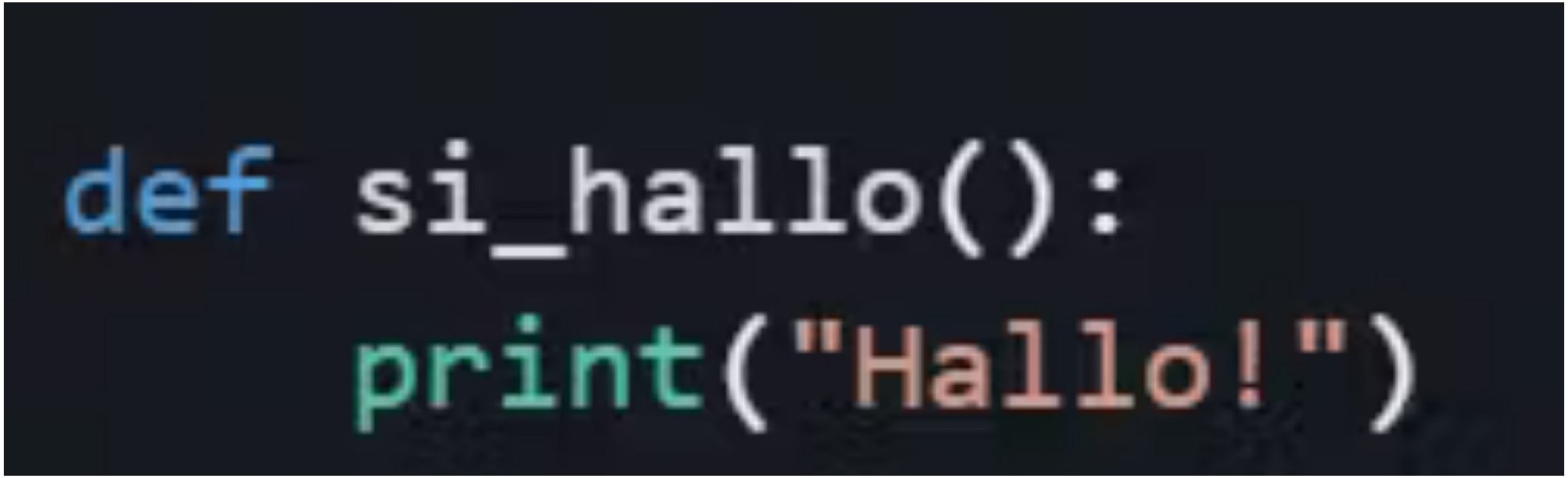
```
6>6
6>=6
6<6
6<=6
6==6
6 ! =6
```

- Hva evaluerer uttrykkene til?
- 6 < 6 ---> False
- 6 >= 6 ---> True
- 6 < 6 --- > False
- 6 <= 6 ---> True
- 6 == 6 ---> True
- 6 != 6 ---> False

Prosedyrer uten returverdi eller parametere

Er særlig nyttig dersom man skal gjøre det
samme flere ganger, unngå å repetere
samme kode.

Syntaks:
def funksjonsnavn() :
<det prosedyren skal gjore>


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

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

si_hallo()
si_hallo()
si_hallo()
```

Feilmeldinger

Det finnes to hovedtyper:

- Syntaksfeil
- Unntak/exceptions

Det finnes også logiske feil!

```
  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'e
```

Finn 3 feil i dette programmet:

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

if tall < 5:
   print(“Tallet er større enn 5”)
#(“mindre enn 5)
else:
   print(“Tallet er ikke større enn 5”)
#(“ikke mindre enn 5”)
```