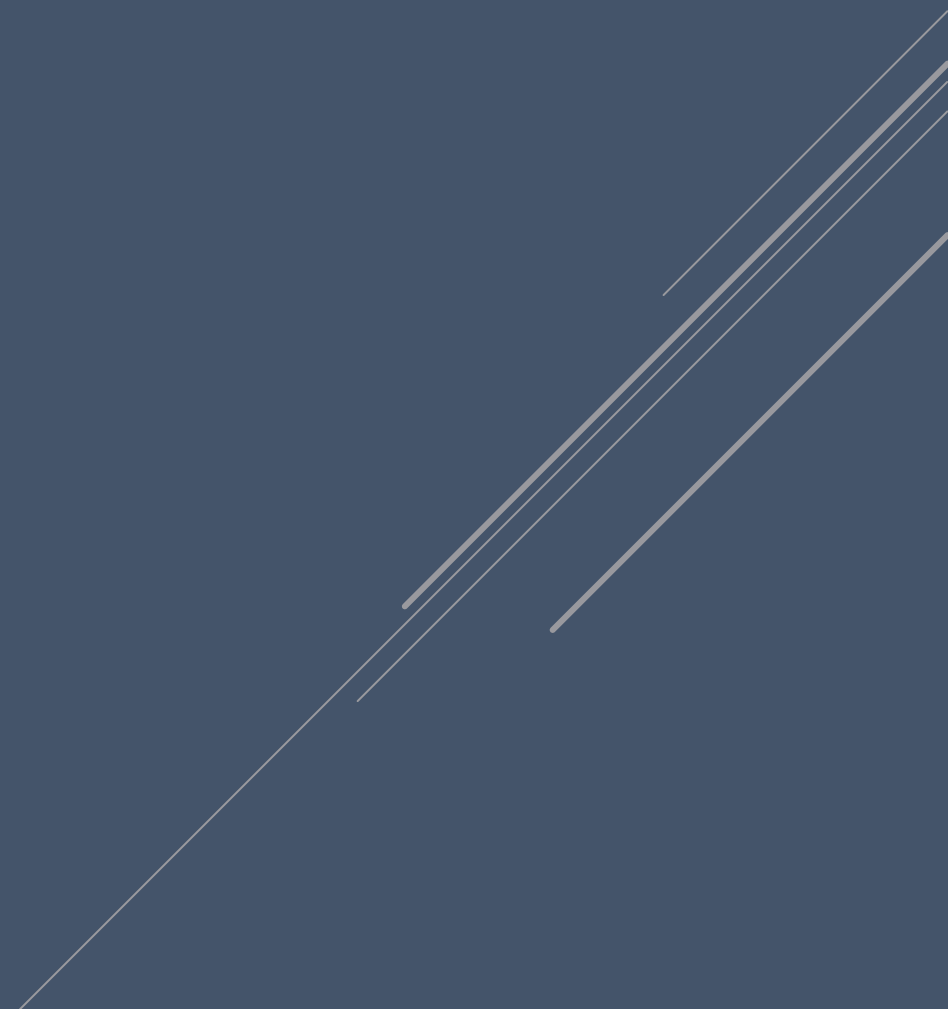



INF100 KRÆSJKURS

Henrik Olsen Eide

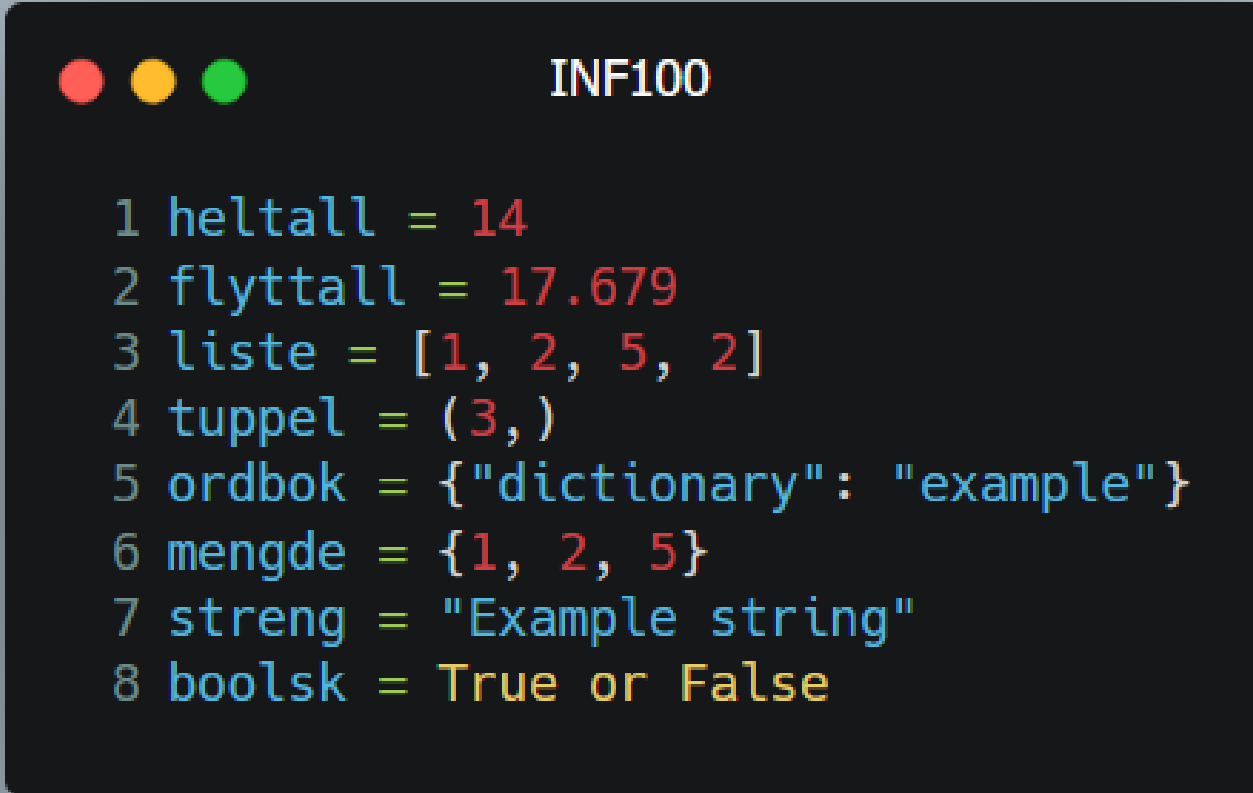


PLAN

- ▶ Typer/datatyper
 - ▶ Funksjoner
 - ▶ Error handling/debugging
 - ▶ Vanlige/hyppige feil
 - ▶ Eksamen
 - ▶ Strategi
 - ▶ Øving frem til eksamen
- 

TYPER

- ▶ Numerisk
 - ▶ Integer
 - ▶ Float
- ▶ Collection/sequence
 - ▶ List
 - ▶ Tuple
 - ▶ Dictionary
 - ▶ Set
- ▶ String
- ▶ Boolean



```
INF100

1 heltall = 14
2 flyttall = 17.679
3 liste = [1, 2, 5, 2]
4 tuppel = (3,)
5 ordbok = {"dictionary": "example"}
6 mengde = {1, 2, 5}
7 streng = "Example string"
8 boolsk = True or False
```

LISTER/STRINGS

- ▶ Essensiell datatype
- ▶ Indeksert
 - ▶ Slicing
 - ▶ [start : stop : step]
- ▶ append/remove
- ▶ insert/pop
- ▶ Addere (+)
- ▶ reverse
- ▶ sort

```
INF100

1 liste = [4, 6, "potet", [2, 4]]
2
3 print(liste[-1])
4 print(liste[2:4:1])
5
6 liste.append([7, 11])
7 liste.pop(-2)
8 print(liste)
9
10 nyliste = liste.remove("potet")
11 print(nyliste)
12
13 print(liste + ["litt", "ekstra"])
14 print(liste)
15
16 liste.sort()
17 print(liste)
```

LISTER MED OUTPUT

```
INF100

1 liste = [4, 6, "potet", [2, 4]]
2
3 print(liste[-1])
4 print(liste[2:4:1])
5
6 liste.append([7,11])
7 liste.pop(-2)
8 print(liste)
9
10 nyliste = liste.remove("potet")
11 print(nyliste)
12
13 print(liste + ["litt", "ekstra"])
14 print(liste)
15
16 liste.sort()
17 print(liste)
```

```
INF100

1
2
3 [2, 4]
4 ['potet', [2, 4]]
5
6
7 [4, 6, 'potet', [7, 11]]
8
9
10 None
11
12
13 [4, 6, [7, 11], 'litt', 'ekstra']
14 [4, 6, [7, 11]]
15
16
17 TypeError: '<'
```

LIST COMPREHENSION

- ▶ `[expression for item in iterable if condition == True]`
- ▶ Konsis/lesbarhet
- ▶ Struktur
- ▶ Bruk med måte
 - ▶ Unngå rot

```
INF100

1 xs = [43, 78, 56, 90, 47, 9]
2
3 odde_xs = []
4 for x in xs:
5     if x % 2 == 1:
6         odde_xs.append(x)
7
8
9 odde_xs = [x for x in xs if x % 2 == 1]
```

STRINGS

- ▶ Indeksert/slicing
- ▶ lower/upper
- ▶ replace
- ▶ join
- ▶ split



INF100

```
1 print("Tekst meD fORskJellige sTOre BokstAver".lower())
2 tekst med forskjellige store bokstaver
3
4
5 print(" \n Litt vel mye mellomrom her  \n".strip())
6 Litt vel mye mellomrom her
7
8
9 print("Litt vel mye mellomrom fortsatt".replace(" ", ""))
10 Littvelmyemellomromfortsatt
11
12
13 print("".join(["liste", "av", "strings", "til", "en", "string"]))
14 listeavstringstilenstring
15
16
17 print("tilbake til en liste av strings".split())
18 ['tilbake', 'til', 'en', 'liste', 'av', 'strings']
```

F-STRING

- ▶ Formatted strings

```
INF100

1 import math
2
3 navn = "Henrik"
4 alder = 25
5 print(f"My age is {navn} and I am {alder} years old.")
6 "My age is Henrik and I am 25 years old."
7
8 print(f"{navn = } {alder = }")
9 "navn = 'Henrik' alder = 25"
10
11 pi = math.pi
12 print(f"Pi er {pi:.4f}")
13 "Pi er 3.1416"
```


FUNKSJONER

- ▶ Bryter opp strukturen av programmet
- ▶ Samle kode som brukes flere ganger
- ▶ Lesbarhet
- ▶ Gjenbruk
- ▶ Navn/kommentarer
- ▶ return eller print?

```
INF100

1 print(print)
2
3 def addere(x, y):
4     return x + y
5
6 print(addere(5, 8))
7 print(addere("hei", "pådeg"))
8
9 def dividere(x, y):
10    return x/y
11
12 print(addere(5, dividere(10, 2)))
```

INF100

```
1 # Ex 1
2 def dividere(x, y):
3     x/y
4
5 print(dividere(10, 2))
```

INF100

```
1 # Ex 2
2 def dividere(x, y):
3     print(x/y)
4
5 print(dividere(10, 2))
```

INF100

```
1 # Ex 3
2 def dividere(x, y):
3     return x/y
4
5 print(dividere(10, 0))
```

INF100

```
1 # Ex 4
2 def alle_partall(nums):
3     for n in nums:
4         if n % 2 == 0:
5             return n
6
7 xs = [3, 5, 8, 7, 23]
8 print(alle_partall(xs))
```

ERROR HANDLING

- ▶ try/except
- ▶ Unngå å stoppe programmet ved error
- ▶ Om det kan crashe
- ▶ Om du planlegger at det skal crashe

```
INF100

1 def dividere(x, y):
2     return x/y
3
4 try:
5     print(dividere(10, 0))
6 except:
7     print("Error, kan ikke dele på 0!")
8
9
10 try:
11     print(dividere(10, 0))
12 except ZeroDivisionError:
13     print("Error, kan ikke dele på 0!")
14
15
16 try:
17     print(dividere(10, 0))
18 except Exception as e:
19     print(f"Error: {e}")
```

LESE/SKRIVE TIL FIL

- ▶ with open()
- ▶ Pass på hvilken mappe du er i
- ▶ Filnavn må være relative path
 - ▶ Ikke når du leverer!
- ▶ Obs: Ikke overskriv filen

```
INF100

1 with open("fintext.txt", "r") as f:
2     tekst = f.read()
3
4 print(tekst)
5
6
7 with open("fintext.txt", "r") as f:
8     linjer = f.readlines()
9
10 print(linjer)
11
12
13 ny_tekst = "Her er oppdatert text og tall"
14 nye_tall = [12, 89, 41]
15 with open("fintext2.txt", "w") as f:
16     tall_streng = " ".join([str(x) for x in nye_tall])
17     f.write(f"{ny_tekst}\n{tall_streng}")
```


DEBUGGING

- ▶ Sjekk at typene matcher
- ▶ Whitespace
 - ▶ Indentering
- ▶ *print* ofte, det kan være feil i noe du trodde var riktig
- ▶ *assert* for å sjekke om noe er True
- ▶ VSCode debugger
- ▶ Python Tutor steg for steg
- ▶ <https://pythontutor.com/visualize.html#mode=edit>


VANLIGE FEIL PÅ EKSAMEN

- ▶ For tidlig/feilplassert return i løkker
- ▶ print vs return
- ▶ Presedens av operatorer/funksjoner. E.g. $x == 3$ or 4
- ▶ Sammenligne feil typer ting
 - ▶ $xs > 4$, dersom xs er en liste trengs `len()`
- ▶ Nøstede løkker
 - ▶ `[[1, 4, 7], [2, 5, 8], [3, 6, 9]]`

EKSAMEN - ØVING

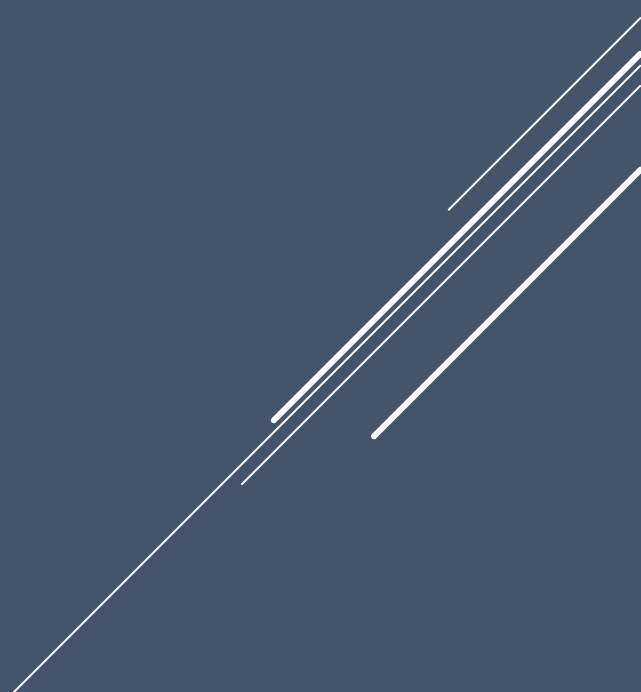
- ▶ Hjemmesiden - inf100.ii.uib.no
 - ▶ Viktige tema
 - ▶ Typer
 - ▶ Funksjoner
 - ▶ Flerdimensjonale lister
 - ▶ Grafikk
- 

STRATEGI


- ▶ Sitter du helt fast så gå videre
 - ▶ Gjør alle oppgavene
 - ▶ Spar pirk/tung debugging til senere
 - ▶ Pass på å få med all info fra oppgaven
 - ▶ Skriv gjerne kommentarer
 - ▶ Ikke vær redd for å teste i fil/REPL, kjør mye kode!
 - ▶ Finn relevante notater/lab'er/dokumentasjon
- 

OPPGAVEGJENNOMGANG

- ▶ Fra prøveeksamen
 - ▶ Etter avstemming ble det oppgave 15



TAKK!

- ▶ Masse lykke til på eksamen!
 - ▶ Hold hodet kaldt
 - ▶ Spør mye teite spørsmål på Discord
- 
- A decorative graphic consisting of several parallel white lines of varying lengths, slanted upwards from left to right, located in the bottom right corner of the slide.